(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 475 554 A2**

(12) **EUROPEAN PATENT APPLICATION**

(21) Application number: **91300904.9**

(51) Int. Cl.^{5}: **H04N 1/46**

(22) Date of filing: **04.02.91**

(30) Priority: **27.12.90 IL 96816
30.12.90 IL 96829
15.01.91 IL 96955
05.02.90 IL 93274**

(43) Date of publication of application:
18.03.92 Bulletin 92/12

(84) Designated Contracting States:
AT BE CH DE DK ES FR GB GR IT LI LU NL SE

(71) Applicant: **SCITEX CORPORATION LTD.
7 Hamada Street
Herzliya 46103(IL)**

(72) Inventor: **Smilansky, Zeev
7 Beeri Street
Rehovot(IL)**
Inventor: **Melman, Haim Zvi
3 Hagai Street
Kfar Saba(IL)**
Inventor: **Bresler, Yoav
24 Refidim Street
Tel Aviv(IL)**
Inventor: **Litvak, Matty J.
42 Ha'atzmaut Street
Petach Tikva(IL)**
Inventor: **Kreitman, Haim
14/3 Hagra Street
Kfar Saba(IL)**
Inventor: **Yad-Shalom, Itai
6 Hagilad Street
Kfar Saba(IL)**

(74) Representative: **Hillier, Peter et al
Reginald W. Barker & Co., 13, Charterhouse Square
London, EC1M 6BA(GB)**

(54) **Apparatus and method for colour calibration.**

(57) There is disclosed technique and apparatus for calibrating a color processing device. The technique includes the steps of comparing a first digital representation of a colored image with a second digital representation thereof and employing at least the transformation to control operation of the color processing device to be calibrated. The first digital representation defines a plurality of first non-scalar color values and the second digital representation defines a plurality of second non-scalar color values corresponding to the plurality of the first non-scalar color values, thereby to provide a transformation pairing each individual one of the first non-scalar color values with a value relatively close to the corresponding one of the second non-scalar color values. The color processing device may be calibrated generally without reference to human aesthetic judgement.

There is additionally provided a method and apparatus for transforming an element of a domain of a first color printing device to an element of a domain of a second color printing device. The method comprises the steps of providing a first transformation from a first digital representation of a colored image to a second digital representation thereof and a second transformation from a third digital representation of a colored image to a fourth digital representation thereof and comparing the first transformation with the second transformation. The second transformation corresponds to the second color printing device, the first transformation corresponds to the first color printing device and the second and fourth digital representations are defined within a single color space.

EP 0 475 554 A2

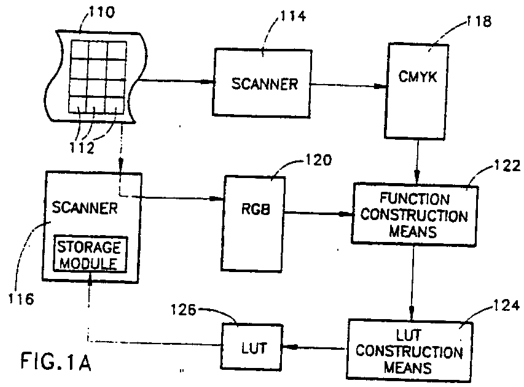

110
112
SCANNER
114
CMYK
118
SCANNER
STORAGE MODULE
116
RGB
120
FUNCTION CONSTRUCTION MEANS
122
LUT
126
LUT CONSTRUCTION MEANS
124

FIG.1A

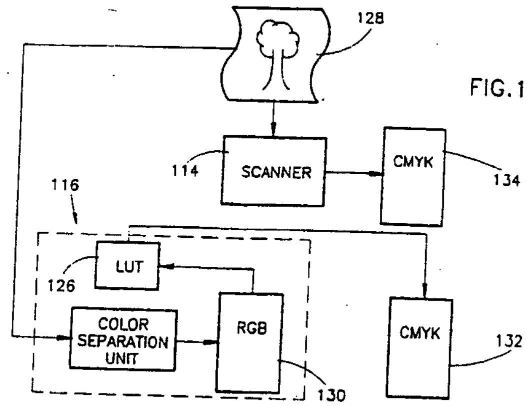

128
SCANNER
114
CMYK
134
116
LUT
126
COLOR SEPARATION UNIT
RGB
130
CMYK
132

FIG.1B

FIELD OF THE INVENTION

The present invention relates to techniques for tone and color reproduction control in graphic arts.

BACKGROUND OF THE INVENTION

Scanning methods are reviewed in R. K. Molla, Electronic Colour Separation, R. K. Printing & Publishing, New York, 1988, the disclosure of which is incorporated herein by reference. The principles of color are explained in G. Wyszecki and W. S. Stiles, Color Science, Wiley & Sons, 1982, the disclosure of which is incorporated herein by reference.

Generally speaking, tone and color reproduction control in high quality graphic arts reproduction is still far from a science. This is particularly evident when a given acceptable result, already realized using given reproduction apparatus, is sought to be realized using other apparatus or using the same apparatus but with a different setting, such as a GCR setting relative to a normal "key black" setting. In such cases, a high degree of expertise, combined with time, effort, expense and patience is required to calibrate the additional apparatus. The results are not always satisfactory.

Unidimensional calibrations in graphic arts, in which a plurality of calibrations are carried out, each being a function of only one color, are known. State of the art techniques include gray balance correction and plotter output calibration techniques. Another example of unidimensional calibration is the automatic TCR (tone and color reproduction) correction process disclosed in published European Application 84307997.1 of Xerox Corporation (Publication number 0144188 A2).

The disadvantage of unidimensional calibrations is that they are only accurate in certain portions of the color space, since a full determination of color is multidimensional, typically having three or four components. For example, the teaching of the above-mentioned published European Application 8437997.1 is relatively inaccurate except in the area of a particular machine's primary color coordinate axes. Gray balance techniques are relatively inaccurate except for a relatively small volume of the color space, comprising gray colors only. Also, the apparatus disclosed in the above-cited published European Application 8437997.1 can be calibrated only by its own output.

Methods of computing a multidimensional function to fit a given set of vectors are known. Interpolative methods may be used if the data is suitably distributed. However the desired conditions regarding the distribution do not always hold in color processing applications, because the data is often not produced directly but rather is the end result of certain procedures (such as scanning, printing, etc.) which are performed on initial preselected data.

An article by Stone et al (Stone, M. C.; Cowan, W. B. and Beatty, J. C., "Color Gamut Mapping and the Printing of Digital Color Images", ACM Transactions on Graphics, 7(4), Oct. 1988, 249-292) discloses use of a colorimeter in mapping a color gamut and printing digital color images, using human aesthetic judgement as a criterion. Human aesthetic judgement is not always the most relevant nor the most efficient criterion for calibrating a color processing device, particularly when it is desired to use another color processing device as a calibration reference. The use of a colorimeter as disclosed by Stone et al is typically cumbersome and time-consuming.

United States Patent 4,500,919 to Schreiber discloses a system for producing color reproductions of an image in which an operator may interactively manipulate a display of the scanned image in order to introduce aesthetic, psychophysically referenced corrections therein. Schreiber teaches that it is desirable for such a system to provide automatic compensation for the effects of ink and paper while allowing the operator to input aesthetic alterations.

United States Patent 4,719,954 to Fujita et al. describes a method and apparatus for creating a color conversion table between scanned colors of a color chart, typically in the Red-Green-Blue (RGB) color coordinate system, and printable colors, typically in the Cyan-Magenta-Yellow-Black (CMYK) color coordinate system, and for using the color conversion table to reproduce a selected measured color in a color specimen. If the selected measured color does not coincide with a value in the color conversion table, an interpolation step is performed.

The method of Fujita et al also includes a correction step when reproduction is performed under different printing conditions. The correction step compensates for the difference between the two printing conditions.

Image creation systems typically comprise a computer with associated graphic software for generating digital representations of color images and/or modifying digital representations of color images, and a plotter or other color output device for transforming the digital representations into analog representations. The analog representation may be created on any suitable substrate, such as on a dia. If desired, e.g. in

pre-press applications, the resulting dia can be scanned.

Examples of commercially available graphic software are Photoshop, by Adobe Systems Inc., Mountainview, CA, USA, usable in conjunction with the Mac II by Apple Computer Inc., USA; and PC Paintbrush Plus, by ZSoft, San Francisco, CA, USA, usable in conjunction with the IBM PC. Examples of commercially available plotters are 4cast, by DuPont, Wilmington, DE, USA, and the LVT Model 1620 digital image recorder by Light Valve Technology, Rochester, NY, USA.

SUMMARY OF THE INVENTION

The following terms as used in the present specification and claims should be construed in the following manner:

Analog representation of a colored image: A representation of a colored image which is perceivable by the human eye as a colored image. The representation may appear upon a transparency, a photograph, a CRT display, a printed page, etc.

Digital representation of a colored image: Any representation of a colored image which is expressed in discrete symbols, such as numerical symbols. A common digital representation of a colored image is a digital file comprising a plurality of numerical values corresponding to a plurality of pixels into which the colored image has been divided, each such numerical value representing some aspect pertaining to the colored appearance of the corresponding pixel.

Substrate: Physical apparatus bearing or displaying an analog representation of an image, e.g. transparency, Cromalin (registered trademark), CRT display, photograph, paper, surfaces suitable for painting on, etc.

Range of color processing apparatus: The totality of color values which can be output by the color processing apparatus.

Domain of color processing apparatus: The totality of color values which can be input by the color processing apparatus.

Color processing apparatus: Apparatus which inputs a first representation of a colored image (digital or analog) and converts it into a second representation thereof (analog or digital), thereby to define a transformation from at least a portion of the range into the domain.

Image creation system: Apparatus which creates an image internally or one which takes as input a representation of a color image and modifies it. Such a system can create the color image from geometrical shapes, can alter the shape and can select and/or modify the color of the color image.

Color reading apparatus: Apparatus which inputs an analog representation of a colored image and converts it to a digital representation thereof. e.g., ECSS, DECSS, colorimeters, spectrum analyzers, densitometers. Typically, the digital representation is expressed in a coordinate system such as XYZ, CMYK, RGB, etc.

Printing machine/device/system; output apparatus, recording apparatus etc.: Any apparatus which inputs a digital representation of a colored image and converts it into an analog representation thereof. For example: conventional, offset, gravure, or other printing apparatus employing inks, conventional or direct digital proofing machines, plotters or color recorders which expose photographic materials, electrostatic printing systems employing powder colorants, color monitors, and color CRT displays.

Calibration: Adjusting color processing apparatus in order to obtain representations, having predetermined substantially objective color characteristics, of colored images sought to be processed.

Color value: A representation of a color, typically in a color coordinate system such as but not limited to RGB, $L^*a^*b^*$, XYZ coordinate systems and device dependent coordinate systems such as color head signals e.g. RGB, ink percentages e.g. CMYK, etc.

Colorant, ink, etc.: Any stimulant of the human eye's light energy receptors, typically through emission, transmission or reflection of photons, including liquid colorants, powder colorants, photographic colorants, phosphors, etc.

Colorant values: A digital representation of the amount of a colorant which it is sought to use.

It is appreciated that any references to color, colored images, color values, colorant values, etc. in the present specification are intended to include the instances of black and white as colors or color values, black and white images, black colorant and ink, etc.

The following abbreviations are used:

TCR:	tone and color reproduction
GCR:	gray component replacement
UCR:	Undercolor removal
UCA:	Undercolor addition
RGB:	red, green, blue. More generally, the term as used herein may refer to any color signals

produced by a color reading device. In a color separation scanner, the term normally refers to the color separation signals of the scanner prior to processing thereof.

CMYK: cyan, magenta, yellow, black (colorants such as inks). More generally, the term as used herein refers to any signals which may serve as input for a color printing device.

ECSS: electronic color separation scanner

DECSS: digital electronic color separation scanner

The present invention seeks to provide a technique for multidimensional calibration of graphic arts reproduction apparatus, which simplifies and greatly expedites the process of calibration of graphic arts reproduction apparatus to faithfully reproduce desired color and tone. Preferably, the technique provides generally accurate calibration of the apparatus throughout substantially the entirety of the range of colors producible by the apparatus.

There is thus provided in accordance with a preferred embodiment of the present invention, a technique for calibrating graphic arts reproduction apparatus using color measuring apparatus (such as the "Smart Scanner" available from Scitex Corporation, Herzlia, Israel, colorimeters, densitometers, etc.) including the steps of providing a transformation of or function from first color values to second color values and employing the transformation to control operation of graphic arts reproduction apparatus. The terms "transformation" and "function" are used interchangeably throughout the present specification. The transformation may be stored as a LUT (look up table) and the data may be transferred using any suitable communication method.

The following procedures, among others, may be greatly simplified and rendered more time efficient and effective using preferred embodiments of the present invention:

1. Incorporating a new color separation scanner (CSS), such as a digital electronic color separation scanner, into an existing reproduction system using automatic calibration that emulates the tone and color reproduction of the existing system.

2. Compensating for a different printing or proofing machine or a different setting on the same machine, by adjustment of the tone and color transformation of a digital electronic color separation scanner, or by adjustment of the digital representation of the picture, such that the printed picture characteristics of tone and color are nearly identical notwithstanding which printing machine or setting is employed.

3. Creating upon a first substrate a duplication of an analog representation of a colored image upon a second substrate. Preferably, both substrates are formed of the same medium or of similar media.

4. Restoring an input copy for given color processing apparatus from an available output copy thereof. Typically, the input and output copies are hard copies. Preferably, the restored input copy, if input to the color processing apparatus, will result in an output copy substantially identical to the available output copy.

5. Incorporating a new digital electronic color separation scanner into an existing reproduction system using automatic calibration to achieve emulation of a UCR (under colorremoval), GCR (gray component replacement) or UCA (under color addition) reproduction produced by the existing system, or to emulate any other special reproduction setting to which the existing system may be set.

6. Calibration of a color monitor display with reference to output apparatus, thereby to provide a representation of a colored image on a color monitor display which is substantially identical to a hard copy representation of that image processed on a given printing device.

7. Enabling an image processing device to process digital data defined in a coordinate system other than the coordinate system of the image processing device.

In accordance with a preferred embodiment of the present invention, the above procedures may be carried out automatically or manually in a straight-forward algorithmic manner, substantially without trial and error. The procedures are preferably non-interactive and are without decision points requiring a decision by a human operator. The procedures may be carried out using an electronic color separation scanner having digital tone and color modules, such as the Smart Scanner, commercially available from Scitex Corporation Ltd., of Herzlia, Israel.

Preferably, the color processing device is calibrated such that substantially all inputs thereto may thereafter be satisfactorily processed thereby.

There is thus provided in accordance with a preferred embodiment of the present invention a technique for calibrating a color processing device including the steps of comparing a first digital representation of a colored image with a second digital representation thereof, the first digital representation defining a plurality of first non-scalar color values, the second digital representation defining a plurality of second non-scalar color values corresponding to the plurality of the first non-scalar color values, thereby to provide a transformation pairing each individual one of the first non-scalar color values with a value relatively close to the corresponding one of the second non-scalar color values, and employing at least the transformation to

control operation of the color processing device to be calibrated, whereby the color processing device may be calibrated generally without reference to human aesthetic judgement.

Further in accordance with a preferred embodiment of the present invention, the technique is algorithmic.

Still further in accordance with a preferred embodiment of the present invention, the technique is non-interactive.

Additionally in accordance with a preferred embodiment of the present invention, the first and second digital representations are not provided by a colorimeter.

Additionally in accordance with a preferred embodiment of the present invention, the first and second digital representations are not in an XYZ coordinate system.

Further in accordance with a preferred embodiment of the present invention, the first and second digital representations are read by color reading apparatus whose colorimetric response differs from the colorimetric response of the human eye.

Still further in accordance with a preferred embodiment of the present invention, the first and second digital representations are read by color reading apparatus whose colorimetric response is not mathematically transformable to any colorimetric response similar to the colorimetric response of the human eye.

Still further in accordance with a preferred embodiment of the present invention, at least some of the non-scalar color values include a black component.

Additionally in accordance with a preferred embodiment of the present invention, the plurality of first non-scalar color values includes a plurality of 3-dimensional color values.

Further in accordance with a preferred embodiment of the present invention, the plurality of second non-scalar color values includes a plurality of 3-dimensional color values.

Still further in accordance with a preferred embodiment of the present invention, the color processing device includes a color reading device or printing device.

Additionally in accordance with a preferred embodiment of the present invention, the technique also includes, prior to the step of comparing, the step of automatically scanning an analog representation of the colored image, thereby to provide at least one of the first and second digital representations.

Still further in accordance with a preferred embodiment of the present invention, the technique also includes the step of using the calibrated device to create upon a second substrate a duplication of an analog representation of a colored image upon a first substrate.

Additionally in accordance with a preferred embodiment of the present invention, the technique also includes the step of using the calibrated device to create an input copy of a colored image which, when processed by the calibrated device, will result in a given output copy of the colored image.

Further in accordance with a preferred embodiment of the present invention, the technique also includes the step of comparing the second digital representation with a third digital representation of the colored image, the third digital representation defining a plurality of third non-scalar color values, thereby to provide a second transformation pairing each individual one of the second non-scalar color values with a value relatively close to the correpsonding one of the third non-scalar color values and the step of employing includes the step of employing both the transformations to control operation of the color processing device to be calibrated.

Additionally in accordance with a preferred embodiment of the present invention, the color processing device to be calibrated includes a color monitor display.

Still further in accordance with a preferred embodiment of the present invention, the first digital representation is characterized in that processing the first digital representation of the colored image with the color processing device provides a second representation of the colored image which defines a provided plurality of color values, each individual one of the provided plurality of color values being substantially equal to a corresponding one of a predetermined plurality of color values falling within the range of the color processing device.

According to another preferred embodiment of the present invention, there is provided apparatus for sampling the color processing characteristics of a color processing device, the color processing device being operative to convert a first representation of a colored image to a second representation thereof, the sampling apparatus including a first representation of a colored image characterized in that processing the first representation of the colored image with the color processing device provides a second representation of the colored image which defines a provided plurality of color values, each individual one of the provided plurality of color values being substantially equal to a corresponding one of a predetermined plurality of color values falling within the range of the color processing device.

Further in accordance with a preferred embodiment of the present invention, the predetermined plurality of color values is characterized in that generally any region of a given size at least partially overlapping the

range of the color processing device contains therewithin at least a predetermined number of color values.

Still further in accordance with a preferred embodiment of the present invention, the first representation includes a digital representation or an analog representation.

Additionally in accordance with a preferred embodiment of the present invention, the second representation includes a digital representation.

According to yet a further preferred embodiment of the present invention, there is provided a method of constructing apparatus for sampling the color processing characteristics of a color processing device, the color processing device being operative to convert a first representation of a colored image to a second representation thereof, the method including the step of repeating at least once the steps of providing first and second representations of a colored image, the representations respectively including a first multiplicity of first color values and a second multiplicity of second color values corresponding thereto, the first and second representations being characterized in that processing the first representation with the color processing device defines the second representation, comparing the first representation of the colored image with the second representation thereof, thereby to provide a transformation, characterized in that operating the transformation on each individual one of the second multiplicity of second color values gives a value substantially equal to the corresponding one of the first multiplicity of first color values, and operating the transformation on the first representation of the colored image, thereby to provide a third representation thereof.

Further in accordance with a preferred embodiment of the present invention, the transformation is a function defined and continuous over generally the entirety of the range of the color processing device.

In accordance with a preferred embodiment of the present invention there is provided apparatus for sampling the color processing characteristics of a color processing device, the sampling apparatus being constructed in accordance with a method for constructing sampling apparatus, the method being in accordance with a preferred embodiment of the present invention.

Further in accordance with a preferred embodiment of the present invention, the step of repeating at least once includes the step of repeating at least twice the steps of providing, processing, comparing and operating and for each repetition from the second onward, the first representation of the colored image provided includes the third representation of the colored image provided in the previous repetition.

Additionally in accordance with a preferred embodiment of the present invention, the third representation defines a plurality of color values and the step of repeating at least twice includes the step of repeating the steps of providing, processing, comparing and operating until the plurality of color values defined by the third representation obtained in the last repetition are at a predetermined degree of closeness to a predetermined plurality of color values located within the range of the color processing device.

There is also provided in accordance with a preferred embodiment of the present invention, a technique for quality control of a color processing device operative to convert a first representation of a colored image to a second representation thereof, the technique including the steps of providing apparatus for sampling the color processing characteristics of the color processing device, the sampling apparatus including a first representation of a colored image characterized in that processing the first representation of the colored image with the color processing device provides a second representation of the colored image which defines a provided plurality of color values, each individual one of the provided plurality of color values being substantially equal to a corresponding one of a predetermined plurality of color values falling within the range of the color processing device, processing the sampling apparatus on the color processing device, and employing the results of the processing step of control oepration of the color processing device.

There is also provided in accordance with yet another preferred embodiment of the present invention a technique for repeatibility testing of a color processing device operative to convert a first representation of a colored image to a second representation thereof, the technique including the steps of providing apparatus for sampling the color processing characteristics of the color processing device, the sampling apparatus including a first representation of a colored image characterized in that processing the first representation of the colored image with the color processing device provides a second representation of the colored image which defines a provided plurality of color values, each individual one of the provided plurality of color values being substantially equal to a corresponding one of a predetermined plurality of color values falling within the range of the color processing device, processing the sampling apparatus on the color processing device; repeating the step of processing on at least one further occasion, and comparing the results of at least two repetitions of the processing steps.

There is provided, in accordance with the present invention, apparatus for transforming an element of a domain of a first color printing device to an element of a domain of a second color printing device and a technique for producing same. The invention is described herein as an apparatus, it being understood that the invention includes a method for performing the operations of the apparatus.

The apparatus includes apparatus for providing a first transformation from a first digital representation of a colored image to a second digital representation thereof and a second transformation from a third digital representation of a colored image to a fourth digital representation thereof and apparatus for comparing the first transformation with the second transformation. The second transformation corresponding to the second color printing device, the first transformation corresponding to the first color printing device and the second and fourth digital representations being defined within a single color space.

Additionally, in accordance with the present invention, the apparatus includes apparatus for storing the output of the apparatus for comparing.

Further, in accordance with the present invention, the output of the apparatus for comparing defines a first plurality of matchings between a first plurality of elements of the domain of the second color printing device and a corresponding first plurality of elements of the domain of the first color printing device and wherein the apparatus for storing include apparatus for providing a second plurality of matchings between the domain of the first color printing device and the domain of the second color printing device, the second plurality exceeding the first plurality in number.

Still further, in accordance with the present invention, the apparatus for storing also includes apparatus for modifying at least some of the corresponding first plurality of elements of the domain of the first color printing device operative prior to the operation of the apparatus for providing a second plurality of matchings.

Moreover, in accordance with the present invention, the apparatus for providing includes apparatus for interpolating between individual ones of the first plurality of matchings, thereby to provide individual ones from among the second plurality of matchings. The apparatus for interpolating includes apparatus for carrying out non-linear interpolation between individual ones of the first plurality of matchings. The apparatus for carrying out non-linear interpolation includes apparatus for fitting a tensor of splines to individual ones of the first plurality of elements of the first domain.

Additionally, in accordance with the present invention, the apparatus of the present invention additionally includes apparatus for employing the stored results to control operation of the first color printing device.

Furthermore, in accordance with the present invention, the apparatus for comparing includes apparatus for searching among the elements of the second digital representation for a plurality of close elements whose values are close to an element of the fourth digital representation, apparatus for forming a multiplicity of subsets, each individual one of the subsets containing some of the plurality of close elements, and apparatus for selecting individual ones from among the multiplicity of subsets. The apparatus for comparing also includes for each individual selected subset, first apparatus for combining the elements of the first digital representation corresponding to the members of the individual selected subset, thereby to define a combination value for the individual selected subset and second apparatus for combining the combination values of each of the individual selected subsets, thereby to provide an element within the first digital representation corresponding to the element within the fourth representation.

Additionally, in accordance with the present invention, the first apparatus for combining includes apparatus for assigning a weight to each member of each individual selected subset, the weights being characterized in that, for each individual selected subset, the element of the fourth digital representation is the center of mass of the weighted members of that subset and apparatus for computing a weighted sum of the members of the individual selected subset. The second apparatus for combining includes apparatus for assigning a weight to each individual selected subset, the weights reflecting the arrangement of the members of the individual selected subset relative to the element of the fourth digital representation and apparatus for computing a weighted sum of the members of the individual selected subset.

Moreover, in accordance with the present invention, the apparatus for selecting includes apparatus for inspecting the arrangements of the members of the multiplicity of subsets relative to the element of the fourth digital representation and apparatus for selecting individual ones from among the multiplicity of subsets in accordance with the result of the step of inspecting.

Still further, in accordance with the present invention, the apparatus includes apparatus for producing a new first transformation using the stored output and apparatus for repeatedly operating the apparatus for comparing and for storing thereby to produce a new transformation between an element of the domain of the first color printing device to an element of the domain of the second color printing device. The apparatus for employing utilize a color converter for converting the domain of the first color printing device to the domain of the second color printing device.

There is also provided, in accordance with the present invention, apparatus for quantifying the appearance of an analog representation of a location of a colored image including apparatus for providing an n-dimensional representation of the location, wherein n is at least 4.

Additionally, in accordance with the present invention, the analog representation of the location is

provided by printing a digital representation of the location and wherein the apparatus for providing includes apparatus for providing a 3 dimensional representation of the color of the location and apparatus for computing at least a fourth value as a function of the digital representation of the location.

Alternatively, the apparatus for computing may be replaced by apparatus for measuring the fourth value directly from the analog representation.

Furthermore, in accordance with the present invention, the three dimensional representation is a colorimetric representation which can be CIE XYZ or CIE L*a*b*.

There is additionally provided, in accordance with the present invention, a system for converting between a digital and an analog representation of an image comprising a translation system for translating a reading and writing point across a substrate, a color proofer comprising the writing point for writing, from the digital representation, the analog representation onto the substrate and a color reading system comprising the reading point for reading the digital representation from the analog representation.

Additionally, in accordance with the present invention, the translation system comprises a drum for rotating the substrate and a translating carriage for translating along the drum.

Moreover, in accordance with the present invention, the system includes apparatus for transforming an element of a domain of a first color printing device to an element of a domain of a second color printing device such as described hereinabove.

There is further provided, in accordance with a preferred embodiment of the present invention, a technique for controlling the operation of an image processing device having a first color coordinate system and including the steps of receiving a digital representation of a color image defined in a second color coordinate system, providing a transformation between the first color coordinate system and the second color coordinate system, employing the transformation to transform the digital representation of the color image into a transformed digital representation of the color image in the first color coordinate system and employing the image processing device to modify the transformed digital representation of the color image.

Additionally, in accordance with a preferred embodiment of the present invention, the digital representation is provided by an input device. The second color coordinate system is the coordinate system of the input device.

Furthermore, in accordance with a preferred embodiment of the present invention, the step of employing the transformation is performed by a device having the second color coordinate system. Alternatively, the step of employing the transformation is performed by the image processing device.

There is further provided, in accordance with a preferred embodiment of the present invention, an image processing device having a first color coordinate system including apparatus for receiving a digital representation of a color image defined in a second color coordinate system, transformation construction apparatus for providing a transformation between the first color coordinate system and the second color coordinate system and apparatus for employing the transformation to transform the digital representation of the color image into a transformed digital representation of the color image in the first color coordinate system. The image processing device is operative to modify the transformed digital representation of the color image.

Additionally, in accordance with a preferred embodiment of the present invention, the digital representation is provided by an input device.

Further, in accordance with a preferred embodiment of the present invention, the second color coordinate system is the coordinate system of the input device.

Finally, in accordance with a preferred embodiment of the present invention, the transformation construction apparatus form part of a device having the second color coordinate system.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be understood and appreciated from the following detailed description, taken in conjunction with the drawings in which:

Figs. 1A and 1B are schematic illustrations respectively of the generation of a calibration transformation or function and its employment in the incorporation of a new color separation scanner or other color reading device into an existing reproduction system employing automatic calibration in accordance with a preferred embodiment of the present invention;

Fig. 2A is a schematic illustration of compensation for a new printing or proofing machine in accordance with a preferred embodiment of the present invention;

Fig. 2B is a schematic illustration of an alternative method of utilizing the calibration information provided by the technique of Fig. 2A;

Figs. 3A, 3B and 3C are schematic illustrations which illustrate respectively a calibration method and two

alternative subsequent uses therefor for generating duplications in accordance with a preferred embodiment of the present invention;

Figs. 4A-4E are schematic illustrations of a technique for restoring an input copy from an output copy in accordance with a preferred embodiment of the present invention;

Figs. 5A and 5B are schematic illustrations of the generation of a calibration transformation or function and its employment in incorporation of a new digital electronic color separation scanner in an existing system for producing UCR, GCR and UCA and any other special setting tone and color reproductions, in accordance with respective alternative embodiments of the present invention;

Fig. 6 is a schematic illustration of a method for calibration of a color monitor display with reference to output apparatus in accordance with a further preferred embodiment of the present invention;

Figs. 7A-7B are schematic illustrations of a reiterative method for providing an improved database for sampling the color processing characteristics of color processing apparatus;

Fig. 8 is an alternative embodiment of a system for performing the method of Figs. 2A and 2B;

Fig. 9 is a block diagram illustration of a technique for transforming a domain of a first color printing device to a domain of a second color printing device, operative in accordance with an alternative embodiment of the present invention;

Fig. 10 is a flow chart illustration of a CMY-CMY conversion technique useful in the technique of Fig. 9;

Fig. 11 is a flow chart illustration of an interpolation technique useful in the embodiment of Fig. 9;

Fig. 12 is a block diagram illustration of a system for reading and writing an image constructed and operative in accordance with the present invention;

Fig. 13 is a general block diagram illustrating apparatus for calibrating a scanner to an image processing unit to a scanner, the apparatus being constructed and operative in accordance with an alternative embodiment of the present invention; and

Fig. 14 is a general block diagram illustrating use of the apparatus of Fig. 13.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

It is an object of the present invention to provide a technique for multidimensional and preferably full range calibration of graphic arts reproduction apparatus, which simplifies and greatly expedites the process of calibration of graphic arts reproduction apparatus to faithfully reproduce desired tone and color. The calibration is accomplished by:

(a) providing a database comprising a set of color values representing a colored image which it is sought to reproduce by using output apparatus employing a plurality of inks or other colorants in amounts defined by the set of color values; and

(b) computing a color calibration function or transformation which fits the database and allows accurate interpolation procedures to be carried out on all color values which are reproducible by the apparatus employed, although only a subset of these values were included in the original database. The function also preferably allows accurate extrapolation procedures to be carried out on color values which fall outside of the convex hull spanned by the color values in the database. Preferably, the function is stored as a look up table (LUT). Alternatively, it can be stored in any other suitable form, such as an analytic formula, etc.

<u>The step of computing a color calibration function</u>

A mathematical formulation of the problem is as follows: given a set of p points defined in a d-dimensional input space and given n sets, each comprising p scalars, to which scalars the points respectively correspond. Each set is contained within a unidimensional space, the n unidimensional spaces corresponding to the n coordinates defining the output space. Find n relatively smooth functions $f_1 ... f_n$ from the d-dimensional space to each of the n unidimensional spaces, such that each of the functions generally fits the data, thereby to allow generally accurate interpolation.

For example: p = 1676, d = 3, n = 4. The p points are 1676 RGB values read by a color reading device, each value being an ordered 3-member set (r, g, b) defining the respective quantities of Red, Green and Blue detected at each of 1676 locations upon a multicolored input image which is to be read. The values r, g and b are scalars and $0 <= r, g, b <= 255$. Each of the p points of the multicolored input image is to be represented by an output image using Cyan, Magenta, Yellow and Black inks (which define four unidimensional spaces), in quantities respectively represented by positive variables c, m, y and k. Four functions, each defined from the three dimensional (r, g, b) space to a respective one of the four one-dimensional c, m, y and k spaces, are to be computed, which will respectively determine the quantity of each of the four

inks which is to be provided to reproduce any particular color (r, g, b).

Any suitable device can be employed to "read" the color values of the database in any of the various embodiments described herein. Inter alia, any of the following devices may be employed: colorimeters, analog electronic color separation scanners, digital electronic color separation scanners, densitometers, spectrum analyzers.

Any suitable color reading device may be used as a calibration reference, such as the Smart Scanner available from Scitex. In some applications, such as Application No. 2 described hereinbelow, it may be desirable to use a scanner whose colorimetric response is similar to that of the human eye, or a scanner whose colorimetric response is mathematically transformable to a response similar to that of a human eye. Any coordinate system suitable for the particular application may be used to represent the color values (XYZ, RGB, etc.). Preferably, substantially all of the colors of the reference color space should be distinguishable by the device used as a calibration reference.

The procedure disclosed hereinbelow is relatively insensitive to the selection of the particular set of p points. In particular, the selected points need not be arranged at regular intervals and do not even need to be distributed homogeneously. However, it is generally preferable to provide a minimum density of data distribution throughout the color space or subspace of interest.

A preferred procedure for computing each function is as follows:

a. Define a set of boxes in the d-dimensional input space. The boxes are preferably selected to be of more than one size, preferably of two different sizes. The set of boxes of each particular size covers the entire d-dimensional space, preferably with overlap. Preferably, the length of the side of each box will be 1/16 to 1/2 of the length of each dimension of the d-dimensional input space, and the boxes will be defined such that the centers thereof are spaced at intervals which are half of the lengths of the sides of the boxes.

For example, and with reference to the numerical example outlined above with eight-bit color values, d=3 and the d-dimensional space is the (r,g,b) cube $[0,255]^3$. In this instance, two sets of boxes may be defined. Each of the 5^3 boxes in the first set is of dimensions 128^3, and each of the 9^3 boxes in the second set is of dimensions 64^3. Boxes of the first set may be defined throughout the cube at intervals of 64 units along each of the three dimensions, beginning at the origin (0,0,0). Boxes of the second set may be defined throughout the cube at intervals of 32 units along each of the three dimensions, beginning at the origin (0,0,0). If a box defined as above is not entirely contained within the d-dimensional space (such as the boxes whose centers are at the origin, only 1/8 of which are contained within the d-dimensional space), the box is defined as the overlap of the box (when defined as above) with the d-dimensional space. For example, the first box in the first set will be $[0,64]^3$.

b. Approximate the transformation within each box by computing a function from that box to the (c,m,y,k) space which optimally fits the data points within that box. Any suitable curve fitting method may be used, such as the methods described in chapter 4 of J. Stoer, Introduction to Numerical Analysis, Springer-Verlag, New York, 1980.

If linear approximation methods are used, it is desirable, in the present example, that there be at least 10 points in each box of dimensions 64^3, including those boxes which are only partially contained within the d-dimensional space, such as the box whose center is at the origin. Therefore, there should preferably be a more dense distribution of points near the periphery of the d-dimensional space.

c. For each of the p points in the input space (or, if the function is stored in LUT form, for each of the LUT points), define the function at that point as a weighted sum of all the approximating functions from all the boxes (some of the weights may be zero). The weight assigned to each approximating function in the weighted sum typically decreases with the size of the corresponding box and with the distance between the center of the corresponding box and the point and increases with the number of points contained within the corresponding box. According to one preferred embodiment of the present invention, a nonanalytical function may be used. According to a further preferred embodiment of the present invention, an analytical formula may be used. One such analytical formula giving the weight for the value of the approximation function f_i derived from box i at a point x_j in box j is:

$$\text{weight}\ (f_i, x_j) = \frac{(\text{number of points in box } i)}{(\text{volume of box } i) \times (\text{distance of point } x_j \text{ from center of box } i)^2}$$

The weights are non-negative. For each x_j, they sum to unity over the totality of boxes of all sizes.

These computations may be performed by any suitable computer or by a suitable digital electronic color separation scanner such as the Smart Scanner available from Scitex. Appendix A is a computer listing of a procedure which inputs a plurality of p d-dimensional color values and a plurality of sets containing p scalars each, and uses the above method to output a color calibration LUT in accordance with a preferred embodiment of the present invention.

Alternative procedures for computing each function $f_1 \ldots f_n$ are:

1. Using a regular grid to order the input or output data, and inverting one of the transforms, as disclosed in the above-referenced article by Stone et al, the disclosure of which is incorporated herein by reference; or

2. Using iterative methods to optimize a sample of the values of the function. Iterative methods are described in chapter 8 of J. Stoer, Introduction to Numerical Analysis, Springer-Verlag, New York, 1980.

It is noted that the above procedures are merely illustrative of the type of mathematical procedures which could be used to compute color calibration functions or transformations having the characteristics specified above.

The step of providing a database

As stated hereinabove, the method of the present invention includes the step of providing a database comprising a representation of a plurality of colors, which database may then be processed by color processing apparatus. To continue the above numerical example, the original database, whose characteristics can be directly controlled by the operator, are the unprocessed RGB values and these values may be recorded, e.g. on a transparency, and then scanned. The data which forms the basis for the function construction procedure described above will then be the processed data, i.e. the transformed form of the original database obtained by processing (recording and scanning) the original database.

It is therefore appreciated that a "good" database for sampling the operation of color processing apparatus over a range or subrange thereof has the property that, once processed by the color processing apparatus, it will, for generally any region of a predetermined size overlapping or contained within the range or subrange, include at least a predetermined number of color values located interiorly of that region. A more general requirement for a "good" database is that, once processed, it is of a predetermined degree of closeness to a "target" comprising a predetermined plurality of color values. However, it is generally the case that if a database which possessed this property prior to being processed is subsequently processed by the color processing apparatus, the processed data will no longer possess the desired property but rather will be "distorted" due to the transformations of the data induced by the color processing procedure.

The color calibration function computation procedure described above can be used in accordance with the method described hereinbelow to produce a "good" database, in the sense set forth hereinabove, from an initial database which may be far from possessing the desired characteristic set forth hereinabove. According to one preferred embodiment, the initial database, prior to being processed by the color processing apparatus, comprises a set of points distributed generally evenly throughout generally the entirety of the domain of the apparatus.

The improvement process of the initial database may, if desired, be continued iteratively until any desired degree of "quality", in the sense described hereinabove, of the final database, is attained. Typically, only three or less such iterations are necessary.

A preferred method in which the function construction procedure described hereinabove is used to provide an improved database will now be described with reference to Figs. 7A-7B, which are schematic illustrations of the steps of the method. Each of steps (a) - (e) of the method is designated in Fig. 7A by the appropriate letter. Step (f) is illustrated in Fig. 7B.

It is appreciated that the method of Figs. 7A-7B may be employed to construct a first database which, once processed by given color processing apparatus, provides a second, processed database which defines a plurality of color values, each of which is substantially equal to a corresponding one of any predetermined "target" plurality of color values physically obtainable by means of the color processing apparatus. The steps illustrated in Fig. 7A are as follows:

(a) Provide an initial digital representation 50 of a colored image, the colored image comprising a plurality of colored locations. The digital representation 50 comprises a plurality of color values such as RGB values, to be referred to as "RGB_0 data", which corresponds to the plurality of colored locations. An example of a preferred initial database 50 useful in deriving a final database for using a Smart Scanner (commercially available from Scitex Corporation, Herzlia, Israel) for sampling the operation of a 4cast color recording device (commercially available from DuPont) is disclosed in Appendix C. The corresponding final database, provided in accordance with the method of Figs. 7A-7B, is disclosed in

Appendix D.

Record the RGB_0 data with a color recording device 56 (such as a 4cast, commercially available from DuPont) to obtain an analog representation 52 of the colored image comprising a plurality of colored locations 54. Preferably, the initial digital representation 50 of the initial colored image will span generally the entirety of the color coordinate system defined by the color recording device.

(b) Read the image 52 using desired input apparatus 58, thereby defining a plurality 60 of color values such as RGB values, to be referred to as "RGB_1 data", which correspond to the plurality of colored locations 54. It is noted that, generally, the RGB_1 data obtained from the original RGB_0 data by recording and scanning will no longer have the same values as the original RGB_0 data. If the values of the RGB_1 data are not sufficiently close to a "target" predetermined plurality of color values, follow steps (c) to (f) below:

(c) Compute a function f_1 from the RGB_1 data 60 to the RGB_0 data 50 by pairing each RGB_1 datum with the value of the corresponding RGB_0 datum and by using the color calibration function construction method described hereinabove. The computation and storage of the color calibration function may be performed by any suitable computation means 62 and storage means 66 respectively. Two software implementations of computation means 62 in conjunction with storage means 66 are disclosed herein as Appendix A (hardware environment: Prisma System workstation, commercially available from Scitex) and Appendix B (hardware environment: Smart Scanner, commercially available from Scitex).

(d) Define RGB_2 data 64 by operating f_1 on each of the RGB_0 values. The RGB_2 data is a digital representation of an "improved" database (relative to the initial RGB_0 database) in the sense described hereinabove with reference to the term "good" database. The operation of function f_1 on the data 50 may be performed by any suitable means 66, such as the Scitex Smart Scanner or any conventional computer such as an IBM PC.

(e) If an analog representation of the improved database is desired, output the RGB_2 file 64, which may be stored by any suitable means, such as the storage module of the Smart Scanner, using color output device 56 as a color printing device and using a substrate 68 of a medium generally identical to the medium of the substrate 52.

(f) If it is desired to continue the above procedure to obtain a still further improved database, i.e. a database whose values are still closer to the "target" predetermined plurality of color values, continue as in Fig. 7B: Provide a digital representation of the output 68 of (e), as in (b), using input apparatus 58, thereby defining a plurality of color values such as RGB values, to be referred to as "RGB'_2 data" 80.

Define a function f_2 from the RGB'_2 data 80 to the RGB_2 data 64 and store it in module 66, as in (c) above.

Define and store RGB_3 data 82 by operating f_2 on each of the RGB_2 values 64, as in (d) above.

If desired, output the RGB_3 data file, as in (e) above. The resulting picture 84 is an analog representation of the still further improved database.

The reiteration or loop of Fig. 7B may be repeated as many times as desired to cause the resultant database to approach the "target" predetermined plurality of color values to any desired degree of closeness.

It is appreciated that the method of providing a data base and the method of computing a color calibration function, both as shown and described hereinabove, have a wide variety of applications when used independently or in conjunction with one another. For example, the method of providing a database described hereinabove is useful not merely for the purpose of computing a color calibration function as described hereinabove, but also in any situation in which it is desired to sample the functioning or the characteristics of color processing apparatus, e.g. in quality control and repeatability test situations such as those presented in the foregoing examples. It is appreciated that the foregoing examples are merely illustrative of possible applications in which it is desired to sample the functioning or the characteristics of color processing apparatus.

Example A: A typical situation in which the method of providing a database as shown and described hereinabove is useful in quality control is that of a printing machine or other output apparatus which is found to produce somewhat varying output as a function of fluctuating environmental factors. A database designed to sample the characteristics of the printing machine, constructed in accordance with the method shown and described hereinabove, may be printed periodically on the printing machine. The database is preferably constructed to sample the printing of colors which are known to be sensitive or problematic when printed on that particular machine. The hard copy is then scanned and a suitable color calibration function is constructed to compensate for any drift which may have occurred relative to a previously defined standard.

Example B: The method of providing a database may also be useful in quality control of color reading apparatus such as scanners. For example, if a scanner is thought to be defective, a database designed to

sample the characteristics of that scanner, constructed in accordance with the method shown and described hereinabove, may be scanned by the putatively defective scanner and the result compared to the results of scanning the same database using results from a scanner known to be properly functional. The database is preferably constructed to sample the scanning of colors which are known to be sensitive or problematic when scanned on that particular scanner.

It is appreciated that the above examples are merely illustrative of possible quality control applications. The term "quality control" is here employed to describe any application in which the quality of performance of color processing apparatus is of interest. More generally, it also applies to any situation in which it is of interest to sample the performance of color processing apparatus.

Example C: A typical situation in which the method of providing a database as shown and described hereinabove is useful in repeatability control is that of a scanner which is suspected of being improperly functional for a certain subregion (or the entire region) of the output space, comprising a plurality of colors. The database provision method shown and described hereinabove may be employed to provide a transparency or other representation which, when scanned, will be mapped onto the subregion in question. This transparency may be used to test the scanner and effect suitable corrective procedures thereupon. It is appreciated that this example is merely illustrative of possible repeatability control applications. The term "repeatability control" or "repeatability testing" is here employed to describe any application in which the repeatability of performance of color processing apparatus over time and/or over changing environmental conditions is of interest.

A number of applications of the method of computing a color calibration function, preferably in conjunction with the method of providing a database, both methods as shown and described hereinabove, will be described in detail herein, however, they are intended to be merely illustrative of the range of possible applications.

It is appreciated that only one or a few embodiments of each of the applications disclosed is described in detail hereinbelow, and that the details of implementation described herein are merely illustrative and by way of example, and that the embodiments described herein may be modified in any suitablemanner. For example, any of the applications herein may be implemented on any suitable computer, such as an IBM PC, by performing the required transformation on the digital output file of any ECSS. Measurements of the database may be carried out automatically, as disclosed herein, or manually, using any suitable equipment such as a spectrum analyzer. The measured data may then be input into the computer either automatically or manually.

Application #1: Calibration of a first color scanner with reference to a second color scanner

Reference is now made to Figs. 1A and 1B which illustrate respectively the generation of a calibration transformation and its employment in the incorporation of a new digital electronic color separation scanner into an existing reproduction system employing automatic calibration in accordance with a preferred embodiment of the present invention.

Conventionally, an existing reproduction work shop that purchases a new electronic color separation scanner (CSS) already owns one or more CSSs. During years of work and interaction with their customers, the shop has developed its own unique tone and color reproduction parameters that characterize the reproductions they produce. The tone and color reproduction parameters may depend on at least the following factors:

The type of originals employed, i.e. the brand and type of transparency or reflective copy;
The color separation scanner employed and its calibration;
The plotting system employed;
The printing system employed; and
Aesthetic considerations.

The introduction of a new ECSS normally changes the tone and color reproduction parameters that are realized. A long and tedious process of adjustment of the new ECSS is normally required, involving numerous adjustments by trial and error. Normally the tone and color reproduction parameters existing prior to introduction of the new ECSS are never fully realized.

In accordance with the present invention, the trial and error techniques currently in use are replaced by a fully- or, if desired, semi-automated well-defined and generally algorithmic technique.

In accordance with a preferred embodiment of the present invention, as illustrated in Fig. 1A, there is provided a substrate, such as a transparency, bearing an analog representation of a colored image 110 which typically comprises a plurality of colored locations 112. Preferably, the colored image will comprise a "good" database constructed in accordance with the database provision method shown and described

hereinabove. Here a "good" database is one which, once scanned by the scanner 114 of Fig. 1A, has a predetermined pattern such as a pattern in which there is a minimum density of data in every area of interest. The predetermined pattern may, for example, be a generally even distribution throughout generally the entirety of the physically producible color space, if it is desired to sample generally the entirety of the color space. Therefore, when constructing the colored image 110 in accordance with the database provision method of Fig. 7, the scanner 114 should preferably be used to scan the colored image 52. Alternatively, scanner 116 can be used.

The colored image 110 is scanned both by an existing ECSS 114 which it is sought to emulate and by the new digital ECSS 116. From the existing ECSS 114 a digital representation 118, comprising color values (preferably CMYK values) each corresponding to an individual one of the locations 112, is obtained. These values relate to the amounts of each colorant to be provided by a printing machine.

From the DECSS 116, a digital representation 120 of the locations 112, comprising color values (preferably RGB values) corresponding to each location 112 is provided. These values correspond to the color separations.

It is appreciated that references to RGB values and CMYK values, etc. throughout the present specification are intended to be examples of suitable color coordinates which can be replaced by any other suitable color coordinates, such as XYZ coordinates. Furthermore, there need not be exactly three input dimensions, or exactly three or four output dimensions. Any suitable number of dimensions may be employed.

Function construction means 122 receives pluralities of corresponding color values 118 and 120. Function construction means 122 is constructed and operative to compute a color calibration function from the color values 120 to the color values 118 in accordance with the color calibration function computation procedure shown and described hereinabove. The color calibration function computed by function construction means 122 is received by LUT construction means 124. LUT construction means 124 is operative to construct a LUT 126 relating the colorant values 118 to the RGB values 120 and to store the LUT 126, such as in the TCR module of the scanner 116. Software implementations of function construction and "on the fly" LUT construction are disclosed in Appendices A and B.

It is appreciated that function construction means and LUT construction means referred to throughout the present specification may be formed as a single unit, e.g. as a computer program which, for each point, computes the function at each point and then stores the point in a LUT "on the fly" before continuing to the next point.

As a result of the foregoing technique, an input of any particular input material to the DECSS will produce DECSS outputs with substantially identical CMYK values as those produced on the existing ECSS from the same input material.

Figure 1B illustrates the reproduction of input material using the existing ECSS 114 as opposed to the calibrated DECSS 116. The DECSS 116 scans the input 128, resulting in a first digital representation 130 thereof, which is then converted by the LUT 126 into a second digital representation 132 thereof, representing the required amounts of each colorant. It is seen that the digital representation 134 of the image 128 resulting from scanning by the ECSS 114 will normally be substantially identical to the output 132, as scanned by the DECSS 116.

Color values of the first digital representation which do not appear in the LUT 126 may be interpolated therefrom, using standard methods, such as those disclosed in chapter 2 of J. Stoer, Introduction to Numerical Analysis, Springer-Verlag, New York, 1980.

It is appreciated that the same or similar interpolation methods may be used in all of the applications of the present invention shown and described subsequently. The interpolation methods are preferably carried out automatically by suitable hardware, such as that commercially available from Zoran Corporation, Santa Clara, CA, USA, or from INMOS Limited, Bristol, UK.

In accordance with the embodiment of Fig. 1A, the ECSS 114 output of color values 118, corresponding to the color patches 112, can be stored as a digital file and can be transmitted to function construction means 122 by any suitable technique, such as via a cable connection, or by employing magnetic tape or other medium.

The above-described technique is not limited to automatic reading of colorant values. These values may be manually read one by one from the scanner. The operator may then input into the function construction means 122, as via a keyboard or via any other suitable input means, a list of RGB values and corresponding colorant values. A suitable software program may then be activated that will create the color calibration function and a look up table (if desired). The LUT may be stored in the memory of the DECSS.

Application #2: Output To Output Calibration

Reference is now made to Fig. 2A which is a schematic illustration of calibration procedures for producing a first printing system output substantially identical to the output from a second printing system.

The embodiment of Fig. 2A is particularly useful in calibrating a proofing machine, used to prepare a single copy of a reproduction for preliminary proofing purposes, to emulate a printing machine which it is intended to use to produce the final reproduction. The state of the art technology, such as the Cromalin (registered trademark) system available from DuPont (U.K) Limited, Hertfordshire, UK, produces a reproduction which may differ substantially from the output of the printing machine that the proofing system is intended to emulate. Consequently, the proof must be evaluated by an expert who can judge the quality thereof while attempting to mentally adjust for the expected discrepancies between proof and eventual printed reproduction. The present invention enables the proofing machine to be accurately and algorithmically calibrated so as to emulate the printing machine.

Since there may exist colors that can be printed by the final printing machine but cannot be printed by the proofing machine using any combination of colorants, it is desirable to choose a proofing machine that is compatible with the printing machine. For example, the Cromalin (registered trade mark) proofing system available from DuPont is generally compatible with offset printing machines. Otherwise, "unprintable" colors may be dealt with using any suitable technique, such as the techniques described in the above-referenced article by Stone et al (particularly pages 275-279 thereof), the disclosure of which is incorporated herein by reference.

A further application is when a printing machine needs to be replaced or when it is desired to add an additional printing machine to an existing workshop. Since the new machine may be of a different brand, type or model than the old machine, it is typically found that printing with the same colorant values on the new machine will produce a color with a different appearance. Therefore, it is generally the case that the new printing machine must be adjusted manually, by a trial and error process, until the reproductions obtained therefrom roughly resemble the reproductions obtained from the existing machine. It is typically impossible to obtain complete concordance between the appearances of the reproductions produced by the first and second machines.

The different appearances obtained from different printing or proofing machines may be the result of at least the following reasons: different colorant materials employed, different technologies employed (offset, gravure, web, Cromalin (registered trade-mark), ink-jet, heat transfer, etc.), dot shape of half-tone film or plates, room temperature, humidity, etc.

Comparison of the results from the respective printing devices is preferably carried out in a CIE (Commission International d'Eclairage) standard color space but may also be carried out in any other suitable color space.

A preferred procedure for using a graphic arts reproduction system comprising a first printing device as a reference in order to calibrate a graphic arts reproduction system comprising a second printing device is the following, described with reference to Fig. 2A:

a. Provide a first database 210 and a second database 212 for the first and second printing devices 214 and 216 respectively. The two databases comprise first and second pluralities of colorant values, preferably CMYK values. Preferably, databases 210 and 212 are "good" databases for sampling the operations of output devices 214 and 216 respectively, in the sense that, once printed by printers 214 and 216 respectively and scanned by the scanner 222, each database has a predetermined pattern such as a pattern in which there is a minimum density of data in every area of interest. The predetermined pattern may, for example, be a generally even distribution throughout generally the entirety of the physically producible color space, if it is desired to sample generally the entirety of the color space.

The two databases are constructed in accordance with the database provision method shown and described hereinabove. When constructing database 210, using the database provision method of Fig. 7, the printer 214 should be used. When constructing database 212, the printer 216 should be used. Preferably, the pluralities of colorant values 210 and 212 include only colorant values that are actually used in reproduction tasks, by printers 214 and 216 respectively.

b. Databases 210 and 212 are printed by printing devices 214 and 216 respectively. The resulting images 218 and 220 respectively are scanned by a color reading device 222 such as the Smart Scanner available from Scitex. The digital representations of images 218 and 220 respectively resulting from the scanning thereof are referenced as 224 and 226. Digital representations 224 and 226 each comprise a plurality of color values, such as RGB values. In some applications it may be desirable to convert the pluralities of RGB values 224 and 226 to corresponding pluralities of CIE XYZ values or values from another suitable coordinate system, using known techniques.

c. Function construction means 228 receives pluralities of corresponding color values 212 and 226 and constructs a color calibration function f_1 from 226 to 212. Function construction means 228 is con-

structed and operative in accordance with the color calibration function computation procedure shown and described hereinabove. The color calibration function f_1 computed by function construction means 228 is stored in storage means 230 and represents the amounts of cyan, magenta, yellow and black inks required to print, using printing device 216, a color to be read as a given RGB value by color reading device 222. Implementations of function construction means 228 and of the storage of the resultant function f_1 as a LUT are disclosed in Appendices A and B.

d. Function f_1 is operated on the RGB_2 values of representation 224, resulting in a plurality 232 of CMYK values. For each CMYK value of database 212, the corresponding C'M'Y'K' value in digital representation 232 represents the amounts of the colorants required to produce, by means of printer 216, a colored location which would be read by color reading device 222 as a value substantially equal to the corresponding RGB value in digital representation 224.

e. Function construction means 228 is also operative to receive pluralites of color values 210 and 232 and to compute a color calibration function from the color values 210 to the color values 232. The color calibration function from digital file 210 to digital file 232 is received by LUT construction means 234 and used to construct and store a LUT 236.

LUT 236 therefore represents the conversions of the amounts of cyan, magenta, yellow and black inks required to print using printing device 216, such that the output will appear to the color reading device 222 to be substantially identical to the RGB values read from the unconverted values of c, m, y and k printed by printing device 214.

According to a preferred embodiment, steps (c) through (e) may be performed twice, once exactly as above and once wherein digital representations 210 and 224 replace digital representations 212 and 226 respectively in step (c) and digital representation 226 replaces digital representation 224 in step (d). The resulting plurality of CMYK values will have twice as many values as it did in the previous embodiment, thereby enhancing the accuracy of LUT 236.

According to a first preferred embodiment, when scanning image representations 218 and 220, the white point is selected to be as close as possible to the white-point CMY values of the corresponding printers 214 and 216 respectively. If the selected white point cannot coincide exactly with the corresponding white-point CMY value, a slightly higher white point is typically selected. All other controls are put on their default setting.

According to an alternative preferred embodiment, the white-point is taken on a blank portion of the white paper or background. All other controls are put on their default setting.

It is appreciated that the most appropriate selection of the white point may vary as a function of the particular application and of the particular graphic arts reproduction system employed.

Preferably, colored image representations 218 and 220 are each automatically scanned, thereby to define a plurality of color values corresponding to a plurality of colored locations into which each image is divided. Any suitable procedure may be employed to accomplish this, which procedure may comprise the steps of: automatically passing from pixel to pixel of the colored image while reading and storing the color values of each pixel, defining a plurality of colored locations each comprising a plurality of pixels, and averaging or otherwise combining the values of at least some of the pixels in each colored location, thereby to define a color value for each colored location. A software implementation of a procedure for automatically scanning an analog representation of a colored image is disclosed in Appendix B.

Once constructed, LUT 236 may be utilized in at least two different ways:

(i) If it is desired to print, on printer 216, an image represented as a digital file originally intended for printing by printer 214 so that its appearance to the color reading device 222 will be substantially as when the digital file is printed on printer 214, the digital file is passed through LUT 236 and the resulting transformed digital file is printed on printer 216. The results of printing the digital file on printer 214 and subsequently reading it using color reading device 222 are substantially identical to the results that would be obtained by printing the transformed file on printer 216 and subsequently reading it using color reading device.

(ii) Reference is made to Fig. 2B which illustrates modification of an RGB-to-CMYK LUT 260 incorporated in a color reading device 223 such as a scanner. LUT 260 is suitable for use in conjunction with printing device 214. It is desired to modify LUT 260 and thereby to obtain a modified RGB-to-CMYK LUT 262 which, when loaded onto scanner 223 and used in conjunction with printing device 216 will result in pictures substantially identical to those produced by scanner 223 loaded with LUT 260 and printing device 214, where the term "substantially identical" implies that pictures produced by the two processes will be "seen" as substantially identical by a scanner.

As shown in Fig. 2B, LUT 236 is operated on the values of LUT 260, transforming each CMYK value intended for printer 214 to a CMYK value suitable for printer 216, thereby to obtain LUT 262. Consequently,

the result 264 of scanning a particular image 265 using scanner 222 loaded with LUT 260 and subsequently printing with printer 214 are substantially the same as the result 266 of scanning the image using scanner 222 loaded with LUT 262 and subsequently printing with printer 216. This implies that a scanned representation of picture 264 will comprise generally the same values as a scanned representation, using the same scanner, of picture 266.

Reference is now made to Fig. 8 which illustrates an alternative embodiment of the output to output calibration.

A digital representation 710 of an image, typically in the CMY color coordinate system but alternatively, in any N-dimensional color coordinate system, is sought to be produced by at least two color reproduction systems, typically a proofer 712 and a printer 714, such that digital representations 720 and 722 of analog representations 716 and 718, respectively, are generally identical. Digital representation 710 is analogous to databases 210 and 212 of Fig. 2A and digital representations 720 and 722 are analogous to digital representations 224 and 226 of Fig. 2A.

Digital representations 720 and 722 are typically produced by a colorimeter or other color measuring system, and are typically in a colorimetric color coordinate system, such as XYZ or $L^*a^*b^*$, or any other measurable N-dimensional color coordinate system. Typically, the three dimensions are the CIE defined coordinate system, such as XYZ or $L^*a^*b^*$, and the remaining dimensions are any which may be desired to be measured, such as non-colorimetric effects seen by a human observer. One such dimension P might be the shininess of each color, or P might be derived from CMYK values using a formula such as:

$$P = K - (C + M + Y)/3 \qquad (1)$$

Formula 1 gives an indication of the amount of black used to produce a given color.

Figs. 8 - 11 illustrating this embodiment indicate that the digital representations are in the CMY and XYZ color coordinate systems. It will be appreciated that the present invention is operative for N-dimensional transformations and that the notation CMY and XYZ is by way of explanation only.

It is known in the art that the two reproduction systems, when input an identical digital representation 710, will generally produce somewhat different analog representations 716 and 718. Since it is generally desired to match the output of one color reproduction system, defined here as the proofer 712, to the output of the other reproduction system, defined here as the printer 714, a color transformation 724 between the CMY color coordinate system of the digital representation 710 to the CMY color coordinate system of the proofer 712 is necessary.

The color transformation 724 typically takes as input an N-dimensional transformation lookup table 742 relating the printable coordinate system of the input digital representation 710 with the printable coordinate system of the proofer 712.

Transformation lookup table 742 can be produced according to the function construction method described hereinabove or, alternatively, in accordance with an alternative function construction method as described hereinbelow with respect to Figs. 9 - 11.

Reference is now made to Fig. 9 which illustrates apparatus for performing the alternative method. An N-dimensional color conversion table 730 is built for the printer 714 which converts between a first input digital representation, typically in CMY, CMYK or any N-dimensional printable color coordinate system, and digital representation 722 of the output analog representation 718, in a measurable color coordinate system, such as XYZ or XYZP. Typically but not necessarily, the number of input dimensions is equivalent to the number of output dimensions. Typically the first input digital representation comprises a "good" database, as described hereinabove.

The conversion table 730 is built as follows: the first input digital representation is sent to printer 714 to be printed. The resultant analog representation 718 is measured, via colorimeters or other measurement devices, and the color value of each printed color is stored in conversion table 730 opposite the input digital value which produced it. The precise details of the production of table 730 are given in the manual for the TRANS/4 color converter, manufactured by Scitex Corporation Ltd. of Herzliya, Israel, incorporated herein by reference and set forth in Appendix E. Alternatively, the conversion table 730 can be produced according to the function construction method described hereinabove.

The first input digital representation typically is organized on a N-dimensional grid.

In accordance with the present invention, an N-dimensional conversion table 732 is built for proofer 712 which converts between a second input digital representation, which can be different from the first input digital representation and typically is in a CMY or CMYK or any N-dimensional printable color coordinate system, and a digital representation of an output analog representation produced without the color transformation 724, in a measurable color coordinate system, such as XYZ. The conversion table 732 is

built as described hereinabove for conversion table 730. The conversion table 732 is not necessarily built using the same grid as that of table 730. It can be built from any grid or from no grid. Its CMY values can be different from the CMY values used to produce conversion table 30 or they can be identical. Typically but not necessarily, the number of input dimensions is equivalent to the number of output dimensions.

Conversion tables 730 and 732 are input into a printer-to-proofer transformer 734, described in more detail hereinbelow with reference to Fig. 10, for transforming a plurality of printer CMY values to a plurality of proofer CMY values where the XYZ values of the proofer CMY values are generally close to the XYZ values of the printer CMY values. An XYZ value of a CMY value is defined as the XYZ measured from the color produced by the color reproduction system when the CMY value is input to it.

It will be appreciated that the output proofer CMY value does not necessarily appear in conversion table 732.

Printer-to-proofer transformer 734 can be utilized for producing a transformation table 736 of proofer CMY values for each of the printer CMY values of conversion table 730, or, alternatively as shown by a dotted arrow, for producing a proofer CMY value upon input of a printer CMY value. Transformation table 736 is on the same grid as conversion table 730.

Transformation table 736 can optionally be stored in a color converter 744, such as the TRANS/4 color converter, for converting a printer CMY value to a proofer CMY value in accordance with table 736. For any printer CMY value not found in table 736, a linear interpolation is performed on table 736 in order to produce the output proofer CMY value.

Alternatively, the transformation table 736 can be input to an optional editor 738 for editing table 736. Editor 738 can be any kind of editor or text processor and is operative to allow an operator to manually correct the table 736, if necessary.

For example, in a CMY color coordinate system using a 'percent (%) dot' scale indicating the percentage of each ink to be used to produce the color, white is denoted by (0,0,0). When the white of the printer 714 is darker than the white of proofer 712, transformer 734 typically produces a proofer CMY white value greater than (0,0,0). This typically produces undesired effects such as the existence of screen dots in the analog output of the proofer 712 where none exist in the analog output of the printer 714.

The above undesired effects can be eliminated by editing table 736, via editor 738, to put a (0,0,0) proofer CMY value for a (0,0,0) printer CMY value. However, it will be appreciated that this produces an incorrect colorimetric (XYZ) value for the white produced by proofer 712.

The output of optional editor 738, an edited table 736, or, in the absence of editor 738, a non-edited table 736, is input to a non-linear interpolator 740 for non-linearly interpolating table 736 thereby to produce transformation table 742 which is denser than transformation table 736. The operation of interpolator 740 is described in more detail hereinbelow with reference to Fig. 11.

Transformation table 742 is an N-dimensional CMY-CMY lookup table which is then stored in color converter 744. For any printer CMY value not found in table 736, a linear interpolation is performed on table 742 in order to produce the output proofer CMY value. It will be appreciated that the linearly interpolated value produced from table 742 is generally more accurate than the linearly interpolated value produced from table 736 since table 742 is denser than table 736.

It will be appreciated that, alternatively, color converter 744 can perform a non-linear interpolation from table 736. The present invention does not illustrate this alternative since cost and speed considerations using current computer technology indicate that the above method is presently more desirable.

Transformation table 742 can be used, as is, or it can be made more accurate as follows:

1) Color transformation apparatus 724 utilizes transformation table 742 on the CMY data listed in table 732 to print analog representation 716 whose colorimetric values are subsequently measured.

2) A new conversion table is produced which is then concatenated to conversion table 732 to produce a new version of conversion table 732.

3) The method of Figs. 9 - 11 is repeated, using the new version of conversion table 732.

Steps 1 - 3 can be repeated any number of times to produce an accurate transformation table 742.

Reference is now made to Fig. 10 which details, in flow chart format, the operation of printer-to-proofer transformer 734.

For each entry in the printer conversion table 730, the operation involves searching the proofer conversion table 732 for entries whose XYZ value is near, by some definition, to the printer XYZ value. The results are stored in a table of close values. This is shown in steps 750 - 764 of Fig. 10.

Specifically, step 754 requires the initialization of the table of close values. This may take the form of defining a closeness threshold less than which indicates closeness, or it may take the form of a maximum number of close values allowed.

In the second case, as the proofer table 732 is searched, the closest values found, and their distances,

are stored. If a closer one is found, the furthest of the stored values is released and the closer one kept. Thus, in step 754, the distances for the initial close values are set to high numbers.

In step 758, the distance between the printer XYZ value and the present proofer XYZ value is calculated, typically as the Euclidean distance. Other suitable distance measures can alternatively be used.

Typically, combinations of the close proofer values found in step 762 are then utilized for the next set of calculations. Combinations must minimally be of one more than the dimension size of the input digital representations. Thus, if the digital representations are in CMY, then there will minimally be four proofer values in each combination. The number of elements in the combination can be predetermined by the operator.

The distance between the printer XYZ value and the combination of proofer XYZ values is calculated as a combination of two elements. The first is the sum of the distances between the individual proofer XYZ values and the printer XYZ value.

The second is an "insideness" measure defining whether or not the printer XYZ value falls within the constellation of proofer XYZ values and can be calculated in a number of ways. For example, a set of linear equations can be solved where equations 2 - 5 below are an exemplary set for use with a combination containing four proofer points:

$$P_x = F_{x1}m_1 + F_{x2}m_2 + F_{x3}m_3 + F_{x4}m_4 \quad (2)$$

$$P_y = F_{y1}m_1 + F_{y2}m_2 + F_{y3}m_3 + F_{y4}m_4 \quad (3)$$

$$P_z = F_{z1}m_1 + F_{z2}m_2 + F_{z3}m_3 + F_{z4}m_4 \quad (4)$$

$$1.0 = m_1 + m_2 + m_3 + m_4 \quad (5)$$

where the m_i are unknown values, P_j indicates the printer X, Y or Z values and F_j indicates the proofer X, Y or Z values.

If the printer XYZ value falls inside the proofer XYZ values, indicated by all positive m_i, the insideness measure is given a small positive value. Otherwise, the insideness measure is defined as the absolute value of the sum of the negative m_i.

The distance between the printer XYZ value and the proofer XYZ values is defined as the combination of the insideness measure and the distance sum, where the combination is typically by multiplication but can be by any other suitable operation.

For all proofer XYZ combinations close to the printer XYZ, the steps 772 - 776 are performed.

Weights are calculated such that the weighted vector sum of the proofer XYZ combination is the printer XYZ. In other words, the printer XYZ is at the 'center of mass' of the proofer XYZ combination. This is calculated in step 772 and involves the solution of a linear set of equations, such as equations 2 - 5.

A preliminary proofer CMY value is then calculated in step 774. Specifically, a weighted vector addition of the proofer CMY values of this combination, found in table 732, is performed using the weights calculated in step 772.

In step 776, the preliminary proofer CMY value of step 774 is assigned a weight which is a function of the distance of the combination to the printer XYZ value as calculated in step 768. Typically, the weight assigned is the distance between the combination and the printer XYZ value divided by the sum of these distances over all combinations close to the printer XYZ value.

The output proofer CMY value which produces a generally identical XYZ value as the printer XYZ value is calculated in step 780 as the weighted sum of all the preliminary proofer CMY values where the weights are those assigned in step 776.

It will be appreciated that alternatively, it is possible to select only the closest proofer combination, calculate its CMY value and use that value as the output proofer CMY value.

Table 736 is a lookup table with the printer CMY value vs. the output proofer CMY value calculated in step 780.

Reference is now made to Fig. 11 which details the non-linear interpolation method of interpolator 740.

A non-linear function, such as a tensor of splines, which defines the relationship between printer CMY values and proofer CMY values is fit to optionally edited conversion table 736 in step 794. Tensors of splines are described in detail in chapter 17 of <u>A Practical Guide to Splines</u>, by Carl De Boor, Vol. 27 of the Applied Mathematical Sciences series published by Springer Verlag, New York, 1978, which is incorporated herein by reference. A further reference is <u>An Introduction to Splines for use in Computer Graphics and Geometric Modeling</u>, by Richard H. Bartels et al., published by Morgan Kaufmann Publishers, Inc., Los

Altos, California, 1987.

Once the number and placement of printer CMY values in transformation table 742 are defined, typically via the operator and typically on a grid used by color converter 744, the non-linear function is used to calculate the proofer CMY value for each printer CMY value in table 742. In this manner, table 742 is produced.

Object code for implementing the operations of transformer 734 and interpolator 740, shown in the flow charts of Figs. 10 and 11, are set forth in Appendices F and G, respectively. Exemplary source code indicating how to interface with the object codes of Appendices F and G is set forth in Appendix H along with instructions as to how to use the source code.

Reference is now made to Fig. 12 which illustrates an integrated system for reading and writing an image, suitable for performing the output to output calibration procedures of Figs. 2A, 2B and 8 - 11. Thus, the system is capable of producing the conversion tables 736 and/or 742 of Fig. 9 and LUTs 236, 260 and 262 of Figs. 2A and 2B.

The system comprises a color proofer 800, for writing an analog representation of an image from a digital representation and a color reading system 802, such as a colorimeter or a spectrophotometer, for creating a digital representation of an image from an analog representation. The color reading system 802 is typically directly connected to the color reproduction system 800. A digital data processor 830 controls the operations of both the color proofer 800 and the color reading system 802.

As is known in the art, color proofer 800 typically comprises a translation system comprising a drum 804 for rotating a substrate 805, such as a piece of paper, upon which will be printed the analog representation of the image, and a translating carriage 807 which moves in one direction as the drum 804 rotates. Onto the translating carriage 807 are attached nozzles 806 for receiving inks in accordance with the digital representation of the image and for effecting the printing of the analog representation of the image onto the substrate 805. A controller 808 simultaneously controls nozzles 806 and the movement of drum 804 and translating carriage 807 in accordance with the digital representation of the image received from data processor 830. Controller 808 also reports information regarding the status of proofer 800 to processor 830.

Any suitable color proofer 800 can be used. An example of which is the Iris color proofer, model 3024, manufactured by Iris Graphics Inc. of Massachusetts, USA.

As is known in the art, color reading system 802 typically comprises a reading head 820, such as a spectrophotometric head, for reading the colors of an analog representation and for producing from them a digital representation of the analog representation and a color reading processor 822 for controlling the operation of the reading head 820 and for receiving its output. Reading head 820 typically comprises a light source and a light detector coupled via an optical system. The elements of the reading head 820 are not shown since they are known in the art. Processor 822 is digitally connected to processor 830 from which it receives instructions and to whom it provides data.

Any suitable color reading system 802 may be used. An example color reading system is the Gretag SPM-700 manufactured by Gretag Ltd. of Regensdorf, Switzerland.

In accordance with the present invention, the reading head 820 is fixed to the translating carriage 807 of the color proofer 800. This physical connection 803 is noted in Fig. 12 by a curved line. The analog representation to be read is placed on drum 804 and reading is effected during the simultaneous rotation of the drum 804 and translation of carriage 807. Preferably, during the reading operation, nozzle 806 is not activated and thus, no writing is performed.

It will be appreciated that the placement of the reading head 820 onto the translating carriage 807 enables reading and writing to be performed in one machine.

Data processor 830 typically comprises a processor 832 for receiving data from color reading system 802 and for transmitting instructions to controller 808 and a processing unit 834 for controlling processor 832. Processing unit 834 is operative to match the output of a second color reproduction system to the output of color proofer 800 as described hereinabove with reference to Figs. 2A, 2B and 8 - 11 and to this end, it controls which operation, the reading or the writing operation, will occur at a given moment.

Processor 832 is typically the processor provided with the color proofer 800, such as the processor provided with the Iris color proofer model 3024 and processing unit 834 is typically a workstation, such as the Whisper workstation manufactured by Scitex Corporation Ltd. of Herzliya, Israel.

The operation of the system of Fig. 12 will now be described with reference to the operations of Figs. 8 - 11, it being understood that the system of Fig. 12 is operative also to perform the operations of Figs. 2A and 2B. Similar reference numerals are used to refer to similar elements:

1) Analog representation 718 from the second color reproduction system (i.e. printer 714) is placed on drum 804 and its color values are read by reading head 820, thereby producing digital representation

722 which is stored in processing unit 834.

2) Analog representation 716 is produced by color proofer 800 using digital representation 710, the representation used to produce analog representation 718.

3) Analog representation 716 remains on drum 804 and its color values are read by reading head 820, thereby producing digital representation 720.

4) Processing unit 834 first produces color conversion tables 730 and 732 from representations 716, 718, 720 and 722 and from them produces the transformation table 742.

Application #3 Duplication of Originals

Reference is now made to the schematic illustrations of Figs. 3A and 3B, which illustrate an embodiment of the invention useful in producing duplications of images existing as hard copies on a particular medium (such as but not limited to a transparency or reflective copy). It is noted that a half-toned printed picture can be duplicated entirely analogously to what will be described herein, except that the picture may be de-screened, using conventional techniques (such as those described in Marquet, M., "Dehalftoning of negatives by optical filtering,", Optica Acta 6, 404-405, 1959; Marquet, M and J. Tsujiuchi, "Interpretation of Particular Aspects of Dehalftoned Images," Optica Acta 8, 267-277, 1961; and Kermisch, D. and P.G. Roetling, "Fourier Spectra of Halftone Screens", J. Opt. Soc. Amer. 65, 716-723, 1975, the disclosure of which is incorporated herein by reference).

A preferred method for providing for duplication of images represented on a given medium is as follows. Steps (a)-(d) are illustrated in Fig. 3A. Step (e) comprises two alternative methods for duplicating a given image once steps (a) - (d) have been carried out, illustrated in Figs. 3B and 3C respectively.

a. Provide a first digital representation 310 of a colored image, typically comprising a first plurality of RGBvalues, using the database provision method shown and described hereinabove. Here a "good" database 310 is one which is suitable for sampling the operation of recorder 312 used in conjunction with recording medium 314 and scanner 316, as explained hereinabove in the section on database construction, and is preferably constructed in accordance with the method of Figs. 7A-7B. Therefore, when using the database provision method of Fig. 7A-7B to construct the database 310, the scanner 316 and the recorder 312 should be used for scanning and recording the initial database.

b. Place a substrate 314 of the desired medium in a color recorder 312 such as a 4cast available from DuPont. According to a preferred embodiment of the present invention, the medium of the substrate 314 is the same as the medium of the original 326 (Fig. 3B) which it is desired to duplicate. Load the color recording apparatus 312 with the digital file 310, thereby to provide an analog representation 315 corresponding to the digital representation 310 of the colored image.

c. Read the analog representation 315 using a color reading device 316 such as an analog ECSS or a DECSS, thereby to obtain a second digital representation 318 of the colored image, preferably comprising a second plurality of RGB values corresponding to the plurality 310 of RGB values.

d. Input digital representations 310 and 318 to function construction means 320, which is operative to construct a function from the plurality of color values 318 to the plurality of color values 310 in accordance with the color calibration function computation procedure described hereinabove. The color calibration function computed by function construction means 320 is received by LUT construction means 322. LUT construction means 322 is operative to construct a LUT 324 relating the RGB values 318 to the RGB values 310 and to store the LUT 324 in the TCR module of the scanner 316. The LUT 324 may now be used as follows:

e. Reference is made to Fig. 3B. Given a substrate 326 (preferably of the same medium as substrate 314) bearing an analog representation of a colored image 327, and when it is sought to duplicate the colored image 327 onto a second substrate 328 (preferably of the same medium as substrate 326), the image 327 is scanned by the scanner 316 whose TCR module contains the LUT 324, thereby to obtain a digital representation 330 of the colored image. The digital representation is then recorded by color recording apparatus 312, thereby to obtain a substantially accurate duplicate 332 of the original colored image 327 on substrate 328.

Alternatively, the colored image 327 may be reproduced as in Fig. 3C. As shown, the image 327 is scanned by the scanner 316 using only the color separation unit 334, thereby to define a digital representation 336, preferably comprising a plurality of RGB values, of image 327. The digital representation 336 is stored in storage means 338. The function constructed by function construction means 320 is stored in any suitable storage 340, such as the memory of a suitable computer, preferably in the form of a LUT. Function operation means 321 then operates the function on digital representation 336 which is read from storage means 338, thereby to provide a modified digital representation 330 of image 327. Digital

representation 330 is then recorded by color recording apparatus 312, thereby to obtain a substantialy accurate duplicate 332 of the original colored image 327 on substrate 328.

If desired, certain of the above steps can be peformed manually. Specifically, the RGB color values of the patches 314 may be manually measured with a color separation scanner and then manually input into function construction means 320, as by a keyboard, instead of being scanned.

According to an alternative embodiment, the image 327 on the substrate 326 is scanned itself to provide digital representation 310 (Fig. 3A). This embodiment is particularly useful in certain applications as it employs precisely those colors required for the duplication of the particular image 327.

Application #4: Reconstruction of Input from Output

Reference is now made to Figs. 4A- 4E, which illustrate a further embodiment of the present invention useful in reconstructing a hard copy produced using a given tone and color reproduction system.

Fig. 4A describes a standard reproduction process of an image on a transparency 412 which is printed as a reflective copy 430. If the original transparency 412 is unavailable, it can be reconstructeed using either the processed digital file 424 or the reflective output 430.

Fig. 4B describes an application in which it is desired to create a single image comprising the tree in picture 412 and the sun in picture 414, and to represent it upon a single substrate, thereby to obtain a single representation 432 (such as a reflective copy) of both the sun and the tree. It may be desired to provide transparencies of the representation of the combined image in which the tree resembles the tree in the original picture 412 and the sun resembles the sun in the original picture 414. Preferably, the medium of the original picture 412 is substantially identical to the medium of the original picture 414.

A preferred method of reconstructing the input copy 412 assuming that LUT 422 and digital file 424 are still available is illustrated in Fig. 4C. First, LUT 422 is inverted, using known methods such as those disclosed on page 267 of the above referenced article by Stone et al, thereby to provide an inverted LUT 434. LUT 434 is then operated on digital file 424, thereby to provide a digital file 436, typically comprising a plurality of RGB values, which values are substantially identical to the plurality of RGB values 418 scanned from the input copy 412 (Fig. 4A). The remainder of the procedure consists of constructing a LUT 437 which, when operated on digital file 436, will result in a digital file 438 which when recorded on a substrate 440 (preferably of the same medium as the original 412) by a recorder 442, will result in an analog representation which has the following property: If scanned by scanner 416, analog representation 440 will provide a digital representation 442 substantially identical to digital file 436 (and digital file 418). Preferably, the analog representation also has the property of appearing to the human eye to have substantially the same tone and color as the original 412.

A preferred method of constructing a LUT 437 with at least the former property and typically both properties has been shown and described hereinabove with reference to Fig. 3A, in which the LUT with the desired properties is referenced as LUT 324.

A preferred method of reconstructing the input copy 412 from the output copy 430 when digital file 424 is not available, whereas the printed picture 430 of fig. 4A is available, is illustrated in Figs. 4D and 4E. As shown, the method comprises providing a database 444, which is preferably a "good" database for sampling the operation of printer 428 in conjunction with scanner 416 and which typically comprises a plurality of CMYK values. The database 444 is printed (e.g. as a reflective copy) by printer 428 and is subsequently scanned by scanner 416, thereby to provide a digital file 450. Alternatively, digital file 450 may be predetermined and database 444 may be constructed therefrom using the database construction method shown and described hereinabove with reference to Figs. 7A-7B. Function construction means 452 receives corresponding pluralities of color values 450 and 444 and constructs a color calibration function from RGB values 450 to CMYK values 444 and preferably stores it in the form of a LUT 454, all in accordance with the function construction procedure shown and described hereinabove.

As shown in Fig. 4E, output copy 430 is scanned by scanner 416 and the resulting digital file 456, typically comprising RGB values, is passed through LUT 454, thereby to provide a digital file 458 preferably comprising a plurality of CMYK values. The plurality 458 of CMYK values, when output by printer 428, will result in a hard copy 460 of the original image which is substantially identical to the hard copy 430. The digital file 458 is substantially identical to digital file 424 of Fig. 4A. Therefore, digital file 458 may be employed to restore the original transparency 412 using the procedure of Fig. 4C.

The color recording apparatus 442 may comprise any suitable color recording apparatus, such as the 4cast plotter available from DuPont.

The computations described hereinabove need not be carried out by the scanner but may alternatively be carried out by any suitable computation means, typically a standard computer such as an IBM PC, which

may communicate with the remainder of the apparatus using any suitable conventional communication method.

Application #5 Calibration of a first color separation scanner with reference to a second color separation scanner on a special setting

The following embodiment of the present invention is useful when it is desired to calibrate a scanner or other color reading device relative to a reference scanner/reading device on a special setting such as but not limited to GCR, UCR, UCA, etc. This embodiment is particularly useful if the operator is relatively unfamiliar with the special setting.

Reference is now made to Fig. 5A which illustrates an embodiment of the present invention useful in incorporating a new DECSS into an existing TCR system comprising a currently used ECSS (or DECSS) on a special setting.

It is appreciated that, by putting the currently used ECSS onto its special setting, a look up table can be constructed which will allow the new DECSS to emulate the existing TCR system, by using the method of Figs. 1A and 1B shown and described hereinabove. However, normally, an unmodified implementation of the look up table is undesirable since operators generally find it difficult to perceive and interpret the special setting CMYK values in the course of subsequent operator controlled tone and color adjustment. Therefore, it is preferable to initially scan the image with a scanner loaded with a "regular" LUT in order to enable the operator to carry out the desired tone and color modifications. Once the modifications have been completed, the modified color values may be converted to the special setting values, thereby to implement the calibration of the scanner to be calibrated with reference to the special setting of the reference scanner.

A preferred procedure for calibrating a first color scanner with reference to a second color scanner on a special setting comprises the following steps:

a. The existing scanner 510 is put onto its normal setting N and an analog representation 512 of a colored image comprising a plurality of colored locations 514 is scanned, thereby to obtain a digital representation 516 comprising a plurality of color values, typically CMYK values, corresponding to the plurality of colored locations 514.

The colored image 512 is preferably a "good" database constructed in accordance with the database provision method shown and described hereinabove. Here a "good" database 512 is one whose values are as close as desired to a "target" predetermined plurality of color values. For example, database 512 may comprise a database which is so constructed that it samples the operation of scanner 510 on its special setting in the subrange in which use of a special rather than normal setting makes a substantial difference. Construction of such a database is explained hereinabove in connection with the database construction method of Figs. 7A-7B.

b. The existing scanner 510 is put onto the desired special setting S and the same colored image is scanned, thereby to obtain a digital representation 518 comprising a plurality of color values, typically CMYK values, corresponding to the plurality of colored locations 514.

c. Digital representations 516 and 518 are input to function construction means 520, which is operative to construct a function from the plurality of color values 516 to the plurality of color values 518 in accordance with the color calibration function computation procedure shown and described hereinabove. The color calibration function computed by function construction means 520 is received by LUT construction means 522. LUT construction means 522 is operative to construct a LUT 524 relating the CMYK values 518 to the CMYK values 516 and to store the LUT 524 in the TCR module of the scanner 510. Implementations of means for constructing a function and storing it as a LUT " on the fly" are disclosed in Appendices A and B.

d. When it is desired to use the new DECSS 526 to scan an input copy 528, the input 528 is scanned with the scanner 526, thereby to obtain a digital representation 530 of the input 528. The RGB (typically) values of digital representation 530 are typically converted using the standard LUT 126 of Fig. 1A, resulting in a second digital representation 532 of input 528, preferably comprising a plurality of CMYK values which are "standard" in that they are familiar to a human operator accustomed to working on a normal setting and thus easily modifiable by the operator.

e. Desired tone and color manipulations may be carried out by a human operator, typically on LUT 126, resulting in modifications of digital representation 532 in subsequent versions thereof.

f. Once the operator has completed the step of manipulating tone and color, LUT 524 is employed to convert each of the normal setting CMYK values of the digital representation 532 to the corresponding special setting CMYK values, resulting in a final digital representation 536 of the input 528, which is substantially identical to the digital representation of input 528 which would result by scanning input 528

with scanner 510 on its special setting and performing the same operator-input tone and color manipulations.

Alternatively, as shown in Fig. 5B, following the execution of tone and color modifications by the operator, the CMYK values of LUT 126 may be converted, thereby to define a converted LUT 538, by using the conversion stored in LUT 524 or by operating the function constructed by function construction means 520 on LUT 126. LUT 538 may be stored in the TCR module of the scanner 526. Digital representation 530 may then be directly converted by LUT 538, preferably on the fly, to provide the final digital representation 536.

It is noted that here as throughout the present specification, the color calibration function whose construction is described hereinabove may be stored in the memory of any suitable commercially available computing means, such as the IBM PC.

Application #6: Calibration of a color monitor display with reference to output apparatus

Reference is now made to Fig. 6 which is a schematic illustration of a method for calibration of a CRT with reference to output apparatus. The objective is to provide an analog representation 610, on a CRT display 616, of a colored image, which representation resembles a hard copy representation 612 of the colored image output from a printing device 214. It may be appreciated that the present method and apparatus are generally similar to the method and apparatus of Fig. 2, where a printing device 216 (rather than a CRT) is calibrated with reference to output apparatus 214. Identical reference numerals are employed to reference identical elements in Figs. 2 and 6 to facilitate understanding of the similarity. The distinguishing elements of the method and apparatus of Fig. 6 will now be discussed.

As shown in Fig. 6, an optical interface 620 is required to enable scanner 222 to receive input from CRT 616. The particular interface required varies according to the scanner 222. For example, the Scitex Smart Scanner may be optically coupled to the screen of the monitor 616 by mechanically disconnecting and removing the color separation head of the scanner from the interior of the scanner while maintaining the electrical wiring connections and placing the head in front of the monitor. Also, the RGB values 210 are preferably displayed one after the other on the CRT screen 616 and are received one at a time by the scanner 222. Synchronization of the scanner with the monitor is required. This procedure is provided on the Smar Scanner available from Scitex.

As in Application #2, it may be desirable to convert pluralities of RGB values 224 and 226 to XYZ values or values from any other suitable coordinate system, using conventional apparatus and techniques, such as those described in P.G. Engeldrum, "Almost Color Mixture Functions", Journal of Imaging Technology, 14(4), August 1988, and in references 2 and 5-7 cited therein. The disclosures of this article and of all references cited therein are incorporated herein by reference. Also, it is appreciated that any suitable color reading device may replace the scanner 222.

If desired, RGB values 226 may be read in CMC form by optical interface 620. The filter arrangement of the normally used color separation head of the Smart Scanner may be replaced by a CMC filter arrangement which emulates the human eye. CMC filter arrangements and methods for constructing them are described in the above referenced article entitled "Almost color mixture functions" as well as in references 2, 5 and 6 thereof. Preferably, the illumination of analog representations 610 and 612 should be such that their respective white areas will be of substantially the same brightness.

Application 7: Calibration of an Input Device to an Image Processing Device

Image creation and processing systems may be used to merge or combine more than one color image into a single color image and\or to modify input color images. For example, it may be desired to create a color image comprising a first portion obtained by photographic techniques and subsequently read into the image processing system, preferably using conventional scanning techniques, and a second portion generated by the graphic software of the image processing system. Alternatively or in addition, it may be desired to use the graphic software of the image processing system to modify a digital representation of a photographic image or of any other color image generated externally of the image processing system.

Unfortunately, the coordinates used to define internally generated images, denoted here as creation system coordinates, are typically different than those used to define externally generated images, denoted here as scanner coordinates. For example, a RGB color value (0,0,10) in scanner coordinates might result in a first input color blue while producing an output of a second color blue in creation system coordinates. However, when combining internally and externally generated images, or when modifying an externally generated image within the graphic software, it is desired to match the two coordinate systems such that

the RGB color value (0,0,10) represents the same color for the scanner input and the image processing unit.

A preferred method and apparatus for achieving one or both of the above objectives is shown and described herein. The apparatus and method shown and described herein have the particular advantage of automatically and generally without resorting to human judgement providing an output from the image processing system which generally resembles the input thereto, apart from the modifications or merge operations performed on the input by the image processing system. In this context, two analog representations resemble one another if a color reading device reading both representations will output substantially the same digital representation for both analog representations.

Reference is now made to Fig. 13 which illustrates calibration of an image processing system comprising a color reading device 932, such as a scanner, means for modifying a digital representation of a color image, such as graphic software 954 (Fig. 14), and a color output device 926, such as a plotter or other color output apparatus. The calibration enables an image read by the scanner 932 to be processed and to be subsequently output by the plotter 926, substantially without causing distortion of the representation of the color image input to the scanner 932, except for the intentional modifications performed thereupon by the image processing system.

As seen in Fig. 13, a digital database 922 is input to, generated in, read from an analog image, or otherwise made available to an image processing unit 924. The digital database 922 generally comprises a plurality of digital color values, defined in the color coordinate system of image processing unit 924.

The database 922 typically comprises a digital file which may be generated by any suitable means, such as via graphic software, or alternatively may be input into the image processing system by any suitable communication method, such as via magnetic tape. Database 922 is output by color output apparatus 926, such as a 4cast plotter, commercially available from DuPont, on any suitable substrate 928, such as a transparency, thereby to provide an analog representation 930 corresponding to the digital representation 922 of the database. Preferably the substrate 928 should be of the same medium as the substrate which it is desired to use to bear the analog images 950 which it is intended to process (Fig. 14). By 'same medium' it is meant the same product. Thus, if it is intended to process using ektachrome substrates produced by Kodak Corporation of the USA, then ektachrome substrates are preferably used for substrate 928.

Analog representation 930 is read by color reading device 932, which may comprise any suitable device such as the Smart Scanner, commercially available from Scitex Corporation, Herzlia, Israel. Scanner 932 produces a digital representation 934 whose color values are defined in the coordinate system of scanner 932.

Digital representation 934 of the database and the original digital representation 922 of the database are both then input into transformation construction apparatus 936. Transformation construction apparatus 936 is operative to construct a transformation from the digital representation 934 to the digital representation 922 and to output this transformation in any suitable form. The transformation is, effectively, a transformation from the color coordinate system of the scanner 932 to that of the image processing unit 924 combined with the output device 926.

Preferably, the transformation is output and stored as a LookUp Table (LUT) 938. The LUT 938 may subsequently be loaded onto scanner 932 or onto suitable computing means, such as an IBM PC or a suitable module that may exist in the image processing unit 924, and may then be used as described hereinbelow with reference to Fig. 14.

Transformation construction apparatus 936 is preferably constructed and operative as described hereinabove.

The plurality of digital color values of database 922 typically represents the color space in which the operator of the image processing apparatus works. It typically has a predetermined pattern, typically comprising patches of color, which typically has at least a minimum density of data (digital color values) in every location of interest. The predetermined pattern may, for example, be a generally even distribution throughout generally the entirety of the producible color space. If certain volumes of the color space are more important to the operator, then the operator may define more color patches in that volume to allow a more accurate represenation of the volume.

Specifically, the operator may produce his desired database 922 as follows:

1. An initial database of patches is produced wherein the colors of the patches have a subset of possible three dimensional eight bit digital color values, where a typical subset is defined as the following combinations of Red (R), Green (G) and Blue (B):

R = n*50
G = m*50

$$B = p^*50 \quad (6)$$

where n, m and p are the entirety of integer values between 0 and 5, inclusive. This produces 216 patches.

2. The initial database is output, to a substrate such as film, by the output device 926, thereby producing an analog output.

3. The operator visually selects on the analog output those patches of the initial database that are included in the color space volume that he wishes to be more accurately represented in his desired database 922 and indicates his selections to image processing unit 924. For example, one selected patch might be the patch that was created by:

$$R = 50, G = 100, B = 150.$$

4. For each selected patch, a plurality of patches are created which are close to it in color. For example, eight more colors might be created which are in the color volume around the selected example color by taking the entirety of combinations of:

$$R = n^*25 + 50$$
$$G = m^*25 + 100$$
$$B = p^*25 + 150 \quad (7)$$

where n, m and p are either 1 or -1.

An alternative method for constructing a database with a desired distribution is as described hereinabove with respect to Figs. 7A and 7B.

Use of the calibrated apparatus of Fig. 13 will now be described, with reference to Fig. 14. There is shown an analog representation 950 of a color image which it is desired to process, using the image processing system of Fig. 13 which comprises graphic software 954 or any other suitable means for producing or modifying representations of color images, including combining or merging means 956 for combining more than one representation of color images into a single representation of a single, combined color image, as well as the scanner 932 and the output device 926.

As shown, the analog representation 950 is read by a color reading device 932, such as a Smart Scanner, commercially available from Scitex Corporation, which is substantially identical to scanner 932 of Fig. 13, and which is loaded with LUT 938 of Fig. 13. Alternatively, the color reading device 932 may not be loaded with LUT 938 nor with any representation of the transformation constructed by transformation construction means 936 of Fig. 13. Instead, the output of color separation unit 952 of the scanner 932, normally in the form of RGB separations or quasi log of RGB separations of the scanner, may be input into transformation operation means 958 Transformation operation means 958 may comprise any suitable computer, such as an IBM PC or a module of the image processing system 924, and is operative to transform the input thereto in accordance with LUT 938 constructed by transformation computation means 936 of Fig. 13.

The transformed output of scanner 932, typically comprising a digital file 960 in the color coordinate system of the image processing system 924, is input to image processing system 924, typically comprising graphic software 954 and/or merging means 956. The digital file 960 may be modified by graphic software 954 and/or it may be merged with another digital image 962 by merging means 956. The merged image may, if desired, be modified by graphics software 954. The final modified and/or merged digital representation 964 is then transformed into an analog image 966, using output device 926 of Fig. 13. The portions of analog image 966 from inital image 950 which did not undergo the modification described hereinabove will be substantially identical in appearance to the initial image 950.

Any suitable method may be employed for merging at least a portion of the digital file 960 with at least a portion of digital image 962. Apparatus and methods for merging color images are described and claimed in Applicant's Israel Application No. 93493, the disclosure of which is incorporated herein by reference.

It will be appreciated by persons skilled in the art that the present invention is not limited to what has been particularly shown and described hereinabove. Rather, the scope of the present invention is defined only by the claims that follow:

```
              name MAKE_LUT
              bss
              public root
root:
              db      02h dup(0)
              public LUT
LUT:
              db      02h dup(0)
              public sk
sk:
              db      04h dup(0)
              public treelength
treelength:
              db      02h dup(0)
              public size1
size1:
              db      02h dup(0)
              public rgb_listlength
rgb_listlength:
              db      02h dup(0)
              public rgb_list
rgb_list:
              db      02h dup(0)
              impure
err:
              dw      00h
              dw      00h
              bss
k_flag:
              db      02h dup(0)
msize:
              db      02h dup(0)
W:
              db      02h dup(0)
```

```
weight_factor:
        db      02h dup(0)
cmyk_list:
        db      01e00h dup(0)
matrix:
        db      080h dup(0)
A:
        db      02h dup(0)
        public weight_vec
weight_vec:
        db      02h dup(0)
        public clut
clut:
        db      02h dup(0)
        public mlut
mlut:
        db      02h dup(0)
        public ylut
ylut:
        db      02h dup(0)
        public klut
klut:
        db      02h dup(0)
        public list
list:
        db      02h dup(0)
        public minthresh
minthresh:
        db      02h dup(0)
        public maxthresh
maxthresh:
        db      02h dup(0)
        public lowlimit
lowlimit:
        db      02h dup(0)
        public highlimit
highlimit:
        db      02h dup(0)
        public min_few
min_few:
        db      02h dup(0)
        public f2_exp
f2_exp:
        db      08h dup(0)
        public grey_dist
```

```
grey_dist:
        db      02h dup(0)
        public grey_factor
grey_factor:
        db      02h dup(0)
minval:
        db      08h dup(0)
        public epsilon
epsilon:
        db      08h dup(0)
g_range:
        db      08h dup(0)
f2:
        db      0200h dup(0)

        impure

        public fmult
fmult:
        db      0f0h
        db      068h
        db      0e3h
        db      088h
        db      0b5h
        db      0f8h
        db      0e4h
        db      03eh
        db      09ah
        db      099h
        db      099h
        db      099h
        db      099h
        db      099h
        db      0b9h
        db      03fh
        db      00h
        db      00h
        db      00h
        db      00h
        db      00h
        db      00h
        db      0f0h
        db      03fh
        db      066h
        db      066h
```

```
        db      066h
        db      066h
        db      066h
        db      066h
        db      0e6h
        db      03fh

        bss

        public endc
endc:
        db      02h dup(0)
        public endm
endm:
        db      02h dup(0)
        public endy
endy:
        db      02h dup(0)
        public endk
endk:
        db      02h dup(0)
        public shadow_contrast_flag
shadow_contrast_flag:
        db      02h dup(0)

        code

        public create_zlut
create_zlut:

        strings

L2:
        db      03ah
        db      073h
        db      064h
        db      03ah
        db      070h
        db      072h
        db      069h
        db      073h
        db      06dh
        db      061h
        db      02fh
        db      066h
```

```
        db      069h
        db      06ch
        db      065h
        db      073h
        db      02fh
        db      064h
        db      074h
        db      073h
        db      02fh
        db      06dh
        db      06bh
        db      06ch
        db      075h
        db      074h
        db      05fh
        db      063h
        db      06fh
        db      06eh
        db      066h
        db      02eh
        db      068h
        db      00h

        code

        strings
L3:
        db      072h
        db      062h
        db      00h

        code

        strings
L5:
        db      020h
        db      0ah
        db      020h
        db      04dh
        db      041h
        db      04bh
```

```
        db      045h
        db      05fh
        db      04ch
        db      055h
        db      054h
        db      020h
        db      065h
        db      072h
        db      072h
        db      06fh
        db      072h
        db      020h
        db      020h
        db      020h
        db      020h
        db      025h
        db      073h
        db      020h
        db      00h

        code


        strings

L6:
        db      020h
        db      0ah
        db      020h
        db      066h
        db      069h
        db      06ch
        db      065h
        db      020h
        db      03ah
        db      073h
        db      064h
        db      03ah
        db      070h
        db      072h
        db      069h
        db      073h
        db      06dh
        db      061h
        db      02fh
```

```
        db      066h
        db      069h
        db      06ch
        db      065h
        db      073h
        db      02fh
        db      064h
        db      074h
        db      073h
        db      02fh
        db      06dh
        db      06bh
        db      06ch
        db      075h
        db      074h
        db      05fh
        db      063h
        db      06fh
        db      06eh
        db      066h
        db      02eh
        db      068h
        db      020h
        db      064h
        db      06fh
        db      065h
        db      073h
        db      020h
        db      06eh
        db      06fh
        db      074h
        db      020h
        db      065h
        db      078h
        db      069h
        db      073h
        db      074h
        db      00h

        code

        strings

L8:
```

```
        db      025h
        db      064h
        db      020h
        db      025h
        db      064h
        db      020h
        db      025h
        db      064h
        db      020h
        db      00h

        code

        strings

L9:
        db      025h
        db      064h
        db      020h
        db      00h

        code

        strings

L10:
        db      025h
        db      064h
        db      020h
        db      00h

        code

        strings

L11:
        db      025h
        db      064h
        db      020h
        db      00h

        code
```

```
                strings
L12:
                db      025h
                db      064h
                db      020h
                db      00h

                code


                strings
L13:
                db      025h
                db      064h
                db      020h
                db      00h

                code


                strings
L14:
                db      025h
                db      064h
                db      020h
                db      025h
                db      064h
                db      020h
                db      025h
                db      064h
                db      020h
                db      025h
                db      064h
                db      00h

                code


                strings
```

```
L15:
          db        025h
          db        06ch
          db        066h
          db        020h
          db        00h

          code

          strings
L16:
          db        025h
          db        064h
          db        020h
          db        025h
          db        064h
          db        020h
          db        025h
          db        064h
          db        020h
          db        025h
          db        064h
          db        020h
          db        00h

          code

          strings
L17:
          db        025h
          db        06ch
          db        066h
          db        020h
          db        025h
          db        06ch
          db        066h
          db        020h
          db        025h
          db        06ch
          db        066h
          db        020h
```

```
        db      025h
        db      06ch
        db      066h
        db      020h
        db      00h

        code

        strings

L18:
        db      025h
        db      064h
        db      020h
        db      025h
        db      064h
        db      020h
        db      025h
        db      064h
        db      020h
        db      025h
        db      064h
        db      00h

        code

        strings

L19:
        db      025h
        db      064h
        db      020h
        db      00h

        code

        strings

L20:
        db      025h
        db      064h
        db      020h
```

```
        db      00h

        code

        strings

L24:
        db      025h
        db      06ch
        db      066h
        db      00h

        code

        strings

L28:
        db      025h
        db      06ch
        db      066h
        db      00h

        code

        strings

L32:
        db      025h
        db      06ch
        db      066h
        db      00h

        code

        strings

L36:
        db      025h
        db      06ch
        db      066h
        db      00h
```

```
                code

                strings
        L37:
                db      025h
                db      064h
                db      020h
                db      025h
                db      064h
                db      020h
                db      025h
                db      064h
                db      020h
                db      025h
                db      064h
                db      00h

                code

                strings
        L38:
                db      025h
                db      064h
                db      00h

                code

                strings
        L40:
                db      0ah
                db      020h
                db      063h
                db      06dh
                db      079h
                db      06bh
                db      028h
                db      030h
                db      02ch
```

```
        db      030h
        db      02ch
        db      030h
        db      029h
        db      020h
        db      03dh
        db      020h
        db      025h
        db      064h
        db      020h
        db      025h
        db      064h
        db      020h
        db      025h
        db      064h
        db      020h
        db      025h
        db      064h
        db      020h
        db      0ah
        db      00h

        code

        strings

L42:
        db      020h
        db      0ah
        db      020h
        db      04dh
        db      041h
        db      04bh
        db      045h
        db      05fh
        db      04ch
        db      055h
        db      054h
        db      020h
        db      065h
        db      072h
        db      072h
        db      06fh
        db      072h
```

```
        db      020h
        db      020h
        db      020h
        db      020h
        db      025h
        db      073h
        db      020h
        db      00h

        code

        strings
L43:
        db      020h
        db      045h
        db      072h
        db      072h
        db      06fh
        db      072h
        db      020h
        db      069h
        db      06eh
        db      020h
        db      06fh
        db      063h
        db      074h
        db      072h
        db      065h
        db      065h
        db      020h
        db      0ah
        db      00h

        code

        strings
L45:
        db      020h
        db      0ah
        db      020h
        db      04dh
```

```
        db      041h
        db      04bh
        db      045h
        db      05fh
        db      04ch
        db      055h
        db      054h
        db      020h
        db      065h
        db      072h
        db      072h
        db      06fh
        db      072h
        db      020h
        db      020h
        db      020h
        db      020h
        db      025h
        db      073h
        db      020h
        db      00h

        code

        strings

L46:
        db      020h
        db      045h
        db      072h
        db      072h
        db      06fh
        db      072h
        db      020h
        db      069h
        db      06eh
        db      020h
        db      063h
        db      06fh
        db      06eh
        db      073h
        db      074h
        db      072h
        db      075h
```

```
            db      063h
            db      074h
            db      05fh
            db      06ch
            db      075h
            db      074h
            db      020h
            db      0ah
            db      00h

            code

            linkage

L10001:
            db      09ah
            db      099h
            db      099h
            db      099h
            db      099h
            db      099h
            db      0d9h
            db      03fh

            code

            push    si
            push    di
            enter   05eh, 00h
            mov     word ptr [bp-052h], 00h
            push    offset L3
            push    offset L2
            call    fopen
            add     sp, 04h
            mov     word ptr [bp-05eh], ax
            or      ax, ax
            jne     word ptr L4
            push    offset L6
            push    offset L5
L20006:
            call    printf
            add     sp, 04h
            mov     word ptr [bp-052h], -01h
            jmp     word ptr L7
```

```
L4:
          lea       ax, word ptr [bp-05ch]
          push      ax
          lea       ax, word ptr [bp-05ah]
          push      ax
          lea       ax, word ptr [bp-058h]
          push      ax
          push      offset L8
          push      word ptr [bp-05eh]
          push      050h
          lea       ax, word ptr [bp-050h]
          push      ax
          call      fgets
          add       sp, 06h
          push      ax
          call      sscanf
          add       sp, 0ah
          push      offset minthresh
          push      offset L9
          push      word ptr [bp-05eh]
          push      050h
          lea       ax, word ptr [bp-050h]
          push      ax
          call      fgets
          add       sp, 06h
          push      ax
          call      sscanf
          add       sp, 06h
          push      offset maxthresh
          push      offset L10
          push      word ptr [bp-05eh]
          push      050h
          lea       ax, word ptr [bp-050h]
          push      ax
          call      fgets
          add       sp, 06h
          push      ax
          call      sscanf
          add       sp, 06h
          push      offset lowlimit
          push      offset L11
          push      word ptr [bp-05eh]
          push      050h
          lea       ax, word ptr [bp-050h]
          push      ax
```

```
call    fgets
add     sp, 06h
push    ax
call    sscanf
add     sp, 06h
push    offset highlimit
push    offset L12
push    word ptr [bp-05eh]
push    050h
lea     ax, word ptr [bp-050h]
push    ax
call    fgets
add     sp, 06h
push    ax
call    sscanf
add     sp, 06h
push    offset min_few
push    offset L13
push    word ptr [bp-05eh]
push    050h
lea     ax, word ptr [bp-050h]
push    ax
call    fgets
add     sp, 06h
push    ax
call    sscanf
add     sp, 06h
push    offset minval+06h
push    offset minval+04h
push    offset minval+02h
push    offset minval
push    offset L14
push    word ptr [bp-05eh]
push    050h
lea     ax, word ptr [bp-050h]
push    ax
call    fgets
add     sp, 06h
push    ax
call    sscanf
add     sp, 0ch
push    offset epsilon
push    offset L15
push    word ptr [bp-05eh]
push    050h
```

```
lea     ax, word ptr [bp-050h]
push    ax
call    fgets
add     sp, 06h
push    ax
call    sscanf
add     sp, 06h
push    offset g_range+06h
push    offset g_range+04h
push    offset g_range+02h
push    offset g_range
push    offset L16
push    word ptr [bp-05eh]
push    050h
lea     ax, word ptr [bp-050h]
push    ax
call    fgets
add     sp, 06h
push    ax
call    sscanf
add     sp, 0ch
push    offset fmult+018h
push    offset fmult+010h
push    offset fmult+08h
push    offset fmult
push    offset L17
push    word ptr [bp-05eh]
push    050h
lea     ax, word ptr [bp-050h]
push    ax
call    fgets
add     sp, 06h
push    ax
call    sscanf
add     sp, 0ch
push    offset f2_exp+06h
push    offset f2_exp+04h
push    offset f2_exp+02h
push    offset f2_exp
push    offset L18
push    word ptr [bp-05eh]
push    050h
lea     ax, word ptr [bp-050h]
push    ax
call    fgets
```

```
            add     sp, 06h
            push    ax
            call    sscanf
            add     sp, 0ch
            push    offset grey_dist
            push    offset L19
            push    word ptr [bp-05eh]
            push    050h
            lea     ax, word ptr [bp-050h]
            push    ax
            call    fgets
            add     sp, 06h
            push    ax
            call    sscanf
            add     sp, 06h
            push    offset grey_factor
            push    offset L20
            push    word ptr [bp-05eh]
            push    050h
            lea     ax, word ptr [bp-050h]
            push    ax
            call    fgets
            add     sp, 06h
            push    ax
            call    sscanf
            add     sp, 06h
            mov     ax, word ptr g_range
            sar     ax, 04h
            mov     word ptr [bp-056h], ax
            mov     word ptr [bp-054h], 00h
    L23:
            mov     ax, word ptr [bp-054h]
            cmp     ax, word ptr [bp-056h]
            jge     word ptr L21
            sal     ax, 05h
            add     ax, offset f2
            push    ax
            push    offset L24
            push    word ptr [bp-05eh]
            call    fscanf
            add     sp, 06h
            inc     word ptr [bp-054h]
            jmp     word ptr L23
    L21:
            push    word ptr [bp-05eh]
```

```
          push    050h
          lea     ax, word ptr [bp-050h]
          push    ax
          call    fgets
          add     sp, 06h
          mov     ax, word ptr g_range+02h
          sar     ax, 04h
          mov     word ptr [bp-056h], ax
          mov     word ptr [bp-054h], 00h
L27:
          mov     ax, word ptr [bp-054h]
          cmp     ax, word ptr [bp-056h]
          jge     word ptr L25
          sal     ax, 05h
          add     ax, offset f2+08h
          push    ax
          push    offset L28
          push    word ptr [bp-05eh]
          call    fscanf
          add     sp, 06h
          inc     word ptr [bp-054h]
          jmp     word ptr L27
L25:
          push    word ptr [bp-05eh]
          push    050h
          lea     ax, word ptr [bp-050h]
          push    ax
          call    fgets
          add     sp, 06h
          mov     ax, word ptr g_range+04h
          sar     ax, 04h
          mov     word ptr [bp-056h], ax
          mov     word ptr [bp-054h], 00h
L31:
          mov     ax, word ptr [bp-054h]
          cmp     ax, word ptr [bp-056h]
          jge     word ptr L29
          sal     ax, 05h
          add     ax, offset f2+010h
          push    ax
          push    offset L32
          push    word ptr [bp-05eh]
          call    fscanf
          add     sp, 06h
          inc     word ptr [bp-054h]
```

```
            jmp     word ptr L31
L29:
            push    word ptr [bp-05eh]
            push    050h
            lea     ax, word ptr [bp-050h]
            push    ax
            call    fgets
            add     sp, 06h
            mov     ax, word ptr g_range+06h
            sar     ax, 04h
            mov     word ptr [bp-056h], ax
            mov     word ptr [bp-054h], 00h
L35:
            mov     ax, word ptr [bp-054h]
            cmp     ax, word ptr [bp-056h]
            jge     word ptr L33
            sal     ax, 05h
            add     ax, offset f2+018h
            push    ax
            push    offset L36
            push    word ptr [bp-05eh]
            call    fscanf
            add     sp, 06h
            inc     word ptr [bp-054h]
            jmp     word ptr L35
L33:
            push    word ptr [bp-05eh]
            push    050h
            lea     ax, word ptr [bp-050h]
            push    ax
            call    fgets
            add     sp, 06h
            push    offset endk
            push    offset endy
            push    offset endm
            push    offset endc
            push    offset L37
            push    word ptr [bp-05eh]
            push    050h
            lea     ax, word ptr [bp-050h]
            push    ax
            call    fgets
            add     sp, 06h
            push    ax
            call    sscanf
```

```
        add     sp, 0ch
        push    offset shadow_contrast_flag
        push    offset L38
        push    word ptr [bp-05eh]
        push    050h
        lea     ax, word ptr [bp-050h]
        push    ax
        call    fgets
        add     sp, 06h
        push    ax
        call    sscanf
        add     sp, 06h
        cmp     word ptr shadow_contrast_flag, 00h
        je      word ptr L39
        push    word ptr endk
        push    word ptr endy
        push    word ptr endm
        push    word ptr endc
        push    offset L40
        call    printf
        add     sp, 0ah
L39:
        push    word ptr [bp-05eh]
        push    050h
        lea     ax, word ptr [bp-050h]
        push    ax
        call    fgets
        add     sp, 06h
        push    word ptr [bp-05eh]
        call    fclose
        add     sp, 02h
        call    octree
        mov     word ptr [bp-052h], ax
        cmp     word ptr [bp-052h], 00h
        je      word ptr L41
        push    offset L43
        push    offset L42
        jmp     word ptr L20006
L41:
        call    construct_lut
        mov     word ptr [bp-052h], ax
        cmp     word ptr [bp-052h], 00h
        je      word ptr L44
        push    offset L46
        push    offset L45
```

```
L44:    jmp     word ptr L20006
        mov     ax, 0ffh
        mov     word ptr c_p+022h, ax
        mov     word ptr c_p+020h, ax
        mov     word ptr c_p+01eh, ax
        sub     ax, ax
        mov     word ptr c_p+01ch, ax
        mov     word ptr c_p+01ah, ax
        mov     word ptr c_p+018h, ax
        fldd    cs:word ptr L10001
        fstpf   word ptr exp_num
L7:
        cmp     word ptr [bp-052h], 00h
        jge     word ptr L1
        push    00h
        push    00h
        push    00h
        push    0271ch
        call    handle_exc
        add     sp, 08h
L1:
        fwait
        leave
        pop     di
        pop     si
        ret
        public octree
octree:

        strings
L50:
        db      020h
        db      0ah
        db      020h
        db      00h

        code

        strings
L51:
        db      020h
```

```
        db      045h
        db      06eh
        db      074h
        db      065h
        db      072h
        db      020h
        db      06eh
        db      061h
        db      06dh
        db      065h
        db      020h
        db      06fh
        db      066h
        db      020h
        db      072h
        db      067h
        db      062h
        db      020h
        db      074h
        db      061h
        db      062h
        db      06ch
        db      065h
        db      020h
        db      00h

        code


        strings
L53:
        db      020h
        db      0ah
        db      020h
        db      04dh
        db      041h
        db      04bh
        db      045h
        db      05fh
        db      04ch
        db      055h
        db      054h
        db      020h
        db      065h
```

```
        db      072h
        db      072h
        db      06fh
        db      072h
        db      020h
        db      020h
        db      020h
        db      020h
        db      025h
        db      073h
        db      020h
        db      00h

        code

        strings
L54:
        db      020h
        db      063h
        db      061h
        db      06eh
        db      06eh
        db      06fh
        db      074h
        db      020h
        db      06fh
        db      070h
        db      065h
        db      06eh
        db      020h
        db      074h
        db      061h
        db      062h
        db      06ch
        db      065h
        db      020h
        db      00h

        code

        strings
```

```
L56:
          db       025h
          db       064h
          db       020h
          db       025h
          db       064h
          db       00h

          code


          strings

L58:
          db       020h
          db       0ah
          db       020h
          db       04dh
          db       041h
          db       04bh
          db       045h
          db       05fh
          db       04ch
          db       055h
          db       054h
          db       020h
          db       065h
          db       072h
          db       072h
          db       06fh
          db       072h
          db       020h
          db       020h
          db       020h
          db       020h
          db       025h
          db       073h
          db       020h
          db       00h

          code


          strings
```

```
L59:
        db      020h
        db      06dh
        db      069h
        db      06eh
        db      020h
        db      03ch
        db      020h
        db      030h
        db      020h
        db      07ch
        db      07ch
        db      020h
        db      06dh
        db      061h
        db      078h
        db      020h
        db      03eh
        db      020h
        db      032h
        db      035h
        db      036h
        db      020h
        db      00h

        code

        strings
L60:
        db      025h
        db      064h
        db      00h

        code

        strings
L62:
        db      020h
        db      0ah
        db      020h
        db      04dh
```

```
        db      041h
        db      04bh
        db      045h
        db      05fh
        db      04ch
        db      055h
        db      054h
        db      020h
        db      065h
        db      072h
        db      072h
        db      06fh
        db      072h
        db      020h
        db      020h
        db      020h
        db      020h
        db      025h
        db      073h
        db      020h
        db      00h

        code

        strings

L63:
        db      020h
        db      06fh
        db      075h
        db      074h
        db      064h
        db      069h
        db      06dh
        db      020h
        db      021h
        db      03dh
        db      020h
        db      033h
        db      020h
        db      00h

        code
```

```
        strings

L64:
        db      025h
        db      064h
        db      00h

        code

        strings

L66:
        db      025h
        db      066h
        db      00h

        code

        strings

L67:
        db      025h
        db      064h
        db      00h

        code

        strings

L68:
        db      025h
        db      064h
        db      020h
        db      025h
        db      064h
        db      020h
        db      025h
        db      064h
        db      00h

        code
```

```
        strings

L70:
        db      020h
        db      0ah
        db      020h
        db      04dh
        db      041h
        db      04bh
        db      045h
        db      05fh
        db      04ch
        db      055h
        db      054h
        db      020h
        db      065h
        db      072h
        db      072h
        db      06fh
        db      072h
        db      020h
        db      020h
        db      020h
        db      020h
        db      025h
        db      073h
        db      020h
        db      00h

        code

        strings

L71:
        db      020h
        db      063h
        db      061h
        db      06eh
        db      06eh
        db      06fh
        db      074h
        db      020h
```

```
        db      061h
        db      06ch
        db      06ch
        db      06fh
        db      063h
        db      061h
        db      074h
        db      065h
        db      020h
        db      073h
        db      070h
        db      061h
        db      063h
        db      065h
        db      020h
        db      066h
        db      06fh
        db      072h
        db      020h
        db      06ch
        db      069h
        db      073h
        db      074h
        db      020h
        db      031h
        db      00h

        code

        strings

L74:
        db      020h
        db      0ah
        db      020h
        db      04dh
        db      041h
        db      04bh
        db      045h
        db      05fh
        db      04ch
        db      055h
        db      054h
        db      020h
```

```
        db      065h
        db      072h
        db      072h
        db      06fh
        db      072h
        db      020h
        db      020h
        db      020h
        db      020h
        db      025h
        db      073h
        db      020h
        db      00h

        code

        strings
L75:
        db      020h
        db      063h
        db      061h
        db      06eh
        db      06eh
        db      06fh
        db      074h
        db      020h
        db      072h
        db      065h
        db      061h
        db      064h
        db      020h
        db      074h
        db      068h
        db      065h
        db      020h
        db      06ch
        db      069h
        db      073h
        db      074h
        db      020h
        db      00h

        code
```

```
        strings
L78:
        db      020h
        db      0ah
        db      020h
        db      025h
        db      064h
        db      020h
        db      069h
        db      074h
        db      065h
        db      06dh
        db      073h
        db      020h
        db      072h
        db      065h
        db      061h
        db      064h
        db      020h
        db      00h

        code

        strings
L79:
        db      020h
        db      0ah
        db      020h
        db      04dh
        db      041h
        db      04bh
        db      045h
        db      05fh
        db      04ch
        db      055h
        db      054h
        db      020h
        db      065h
        db      072h
        db      072h
```

```
        db      06fh
        db      072h
        db      020h
        db      020h
        db      020h
        db      020h
        db      025h
        db      073h
        db      020h
        db      00h

        code

        strings

L80:
        db      020h
        db      063h
        db      061h
        db      06eh
        db      06eh
        db      06fh
        db      074h
        db      020h
        db      072h
        db      065h
        db      061h
        db      064h
        db      020h
        db      074h
        db      068h
        db      065h
        db      020h
        db      06ch
        db      069h
        db      073h
        db      074h
        db      020h
        db      00h

        code

        strings
```

```
L82:
        db      020h
        db      0ah
        db      020h
        db      04dh
        db      041h
        db      04bh
        db      045h
        db      05fh
        db      04ch
        db      055h
        db      054h
        db      020h
        db      065h
        db      072h
        db      072h
        db      06fh
        db      072h
        db      020h
        db      020h
        db      020h
        db      020h
        db      025h
        db      073h
        db      020h
        db      00h

        code

        strings

L83:
        db      020h
        db      063h
        db      061h
        db      06eh
        db      06eh
        db      06fh
        db      074h
        db      020h
        db      061h
        db      06ch
        db      06ch
```

```
            db      06fh
            db      063h
            db      061h
            db      074h
            db      065h
            db      020h
            db      073h
            db      070h
            db      061h
            db      063h
            db      065h
            db      020h
            db      066h
            db      06fh
            db      072h
            db      020h
            db      074h
            db      072h
            db      065h
            db      065h
            db      020h
            db      031h
            db      00h

            code

            strings

L88:
            db      020h
            db      0ah
            db      020h
            db      04dh
            db      041h
            db      04bh
            db      045h
            db      05fh
            db      04ch
            db      055h
            db      054h
            db      020h
            db      065h
            db      072h
            db      072h
```

```
                db      06fh
                db      072h
                db      020h
                db      020h
                db      020h
                db      020h
                db      025h
                db      073h
                db      020h
                db      00h

                code

                strings

        L89:
                db      0ah
                db      020h
                db      063h
                db      061h
                db      06eh
                db      06eh
                db      06fh
                db      074h
                db      020h
                db      067h
                db      072h
                db      06fh
                db      077h
                db      020h
                db      074h
                db      068h
                db      065h
                db      020h
                db      074h
                db      072h
                db      065h
                db      065h
                db      020h
                db      00h

                code

                push    si
```

```
          push    di
          enter   064h, 00h
          sub     ax, ax
          mov     word ptr root, ax
          mov     word ptr list, ax
          sub     ax, ax
          mov     word ptr [bp-060h], ax
          mov     word ptr [bp-05eh], ax
          push    word ptr RGB_ptr
          push    050h
          lea     ax, word ptr [bp-052h]
          push    ax
          call    fgets
          add     sp, 06h
          lea     ax, word ptr [bp-05ah]
          push    ax
          lea     ax, word ptr [bp-058h]
          push    ax
          push    offset L56
          push    word ptr RGB_ptr
          push    050h
          lea     ax, word ptr [bp-052h]
          push    ax
          call    fgets
          add     sp, 06h
          push    ax
          call    sscanf
          add     sp, 08h
          cmp     word ptr [bp-058h], 00h
          jl      word ptr L10002
          cmp     word ptr [bp-05ah], 0100h
          jle     word ptr L57
L10002:
          push    offset L59
          push    offset L58
L20025:
          call    printf
          add     sp, 04h
          mov     word ptr [bp-05eh], -01h
          jmp     word ptr L55
L57:
          lea     ax, word ptr [bp-05ch]
          push    ax
          push    offset L60
          push    word ptr RGB_ptr
```

```
        push    050h
        lea     ax, word ptr [bp-052h]
        push    ax
        call    fgets
        add     sp, 06h
        push    ax
        call    sscanf
        add     sp, 06h
        cmp     word ptr [bp-05ch], 03h
        je      word ptr L61
        push    offset L63
        push    offset L62
        jmp     word ptr L20025
L61:
        push    offset size1
        push    offset L64
        push    word ptr RGB_ptr
        push    050h
        lea     ax, word ptr [bp-052h]
        push    ax
        call    fgets
        add     sp, 06h
        push    ax
        call    sscanf
        add     sp, 06h
        cmp     word ptr [bp-05ch], 03h
        jne     word ptr L65
        lea     ax, word ptr [bp-064h]
        push    ax
        push    offset L66
        push    word ptr RGB_ptr
        push    050h
        lea     ax, word ptr [bp-052h]
        push    ax
        call    fgets
        add     sp, 06h
        push    ax
        call    sscanf
        add     sp, 06h
        lea     ax, word ptr [bp-064h]
        push    ax
        push    offset L67
        push    word ptr RGB_ptr
        push    050h
        lea     ax, word ptr [bp-052h]
```

```
            push    ax
            call    fgets
            add     sp, 06h
            push    ax
            call    sscanf
            add     sp, 06h
            push    offset GAIN_LEVEL
            push    offset D_12
            push    offset W_12
            push    offset L68
            push    word ptr RGB_ptr
            push    050h
            lea     ax, word ptr [bp-052h]
            push    ax
            call    fgets
            add     sp, 06h
            push    ax
            call    sscanf
            add     sp, 0ah
L65:
            mov     ax, word ptr size1
            add     ax, 0ah
            mov     word ptr msize, ax
            push    08h
            push    ax
            call    calloc
            add     sp, 04h
            mov     word ptr list, ax
            cmp     word ptr list, 00h
            jne     word ptr L69
            push    offset L71
            push    offset L70
            jmp     word ptr L20025
L69:
            push    00h
            push    00h
            push    03e8h
            push    word ptr RGB_ptr
            call    fseek
            add     sp, 08h
            push    word ptr RGB_ptr
            push    word ptr size1
            push    08h
            mov     ax, word ptr list
            add     ax, 08h
```

```
        push    ax
        call    freadb
        add     sp, 08h
        mov     di, ax
        mov     ax, di
        cwd
        mov     word ptr sk, ax
        mov     word ptr sk+02h, dx
        or      ax, dx
        je      word ptr L76
        push    offset L75
        push    offset L74
        jmp     word ptr L20025
L76:
        push    word ptr RGB_ptr
        call    fclose
        add     sp, 02h
        mov     word ptr RGB_ptr, 00h
        mov     word ptr msize, 0ccbh
        push    014h
        push    word ptr msize
        call    calloc
        add     sp, 04h
        mov     word ptr root, ax
        cmp     word ptr root, 00h
        jne     word ptr L81
        push    offset L83
        push    offset L82
        jmp     word ptr L20025
L81:
        mov     word ptr treelength, 00h
        mov     word ptr [bp-054h], 01h
L86:
        mov     ax, word ptr [bp-054h]
        cmp     ax, word ptr size1
        jg      word ptr L84
        push    ax
        push    word ptr root
        call    growtree
        add     sp, 04h
        or      ax, ax
        jge     word ptr L87
        push    offset L89
        push    offset L88
        jmp     word ptr L20025
```

```
L87:
        mov     ax, word ptr [bp-05ch]
        add     word ptr [bp-054h], ax
        jmp     word ptr L86
L84:
        inc     word ptr treelength
        push    word ptr root
        call    number_of_sons
        add     sp, 02h
L55:
        cmp     word ptr [bp-05eh], 00h
        jge     word ptr L90
        push    00h
        push    00h
        push    00h
        push    0271ch
        call    handle_exc
        add     sp, 08h
L90:
        cmp     word ptr RGB_ptr, 00h
        je      word ptr L91
        push    word ptr RGB_ptr
        call    fclose
        add     sp, 02h
L91:
        mov     ax, word ptr [bp-060h]
        leave
        pop     di
        pop     si
        ret
        public Construct_lut
Construct_lut:

        strings

L95:
        db      020h
        db      0ah
        db      020h
        db      04dh
        db      041h
        db      04bh
        db      045h
        db      05fh
        db      04ch
```

```
        db      055h
        db      054h
        db      020h
        db      065h
        db      072h
        db      072h
        db      06fh
        db      072h
        db      020h
        db      020h
        db      020h
        db      020h
        db      025h
        db      073h
        db      020h
        db      00h
```

code

strings

```
L96:
        db      0ah
        db      020h
        db      063h
        db      061h
        db      06eh
        db      06eh
        db      06fh
        db      074h
        db      020h
        db      061h
        db      06ch
        db      06ch
        db      06fh
        db      063h
        db      061h
        db      074h
        db      065h
        db      020h
        db      073h
        db      070h
        db      061h
        db      063h
```

```
        db      065h
        db      020h
        db      066h
        db      06fh
        db      072h
        db      020h
        db      062h
        db      072h
        db      061h
        db      06eh
        db      063h
        db      068h
        db      020h
        db      00h

        code

        strings
L99:
        db      020h
        db      045h
        db      06eh
        db      074h
        db      065h
        db      072h
        db      020h
        db      06eh
        db      061h
        db      06dh
        db      065h
        db      020h
        db      06fh
        db      066h
        db      020h
        db      069h
        db      06dh
        db      061h
        db      067h
        db      065h
        db      020h
        db      074h
        db      061h
        db      062h
```

```
                db      06ch
                db      065h
                db      020h
                db      00h

                code

                strings
L101:
                db      020h
                db      0ah
                db      020h
                db      04dh
                db      041h
                db      04bh
                db      045h
                db      05fh
                db      04ch
                db      055h
                db      054h
                db      020h
                db      065h
                db      072h
                db      072h
                db      06fh
                db      072h
                db      020h
                db      020h
                db      020h
                db      020h
                db      025h
                db      073h
                db      020h
                db      00h

                code

                strings
L102:
                db      0ah
                db      020h
```

```
        db      063h
        db      061h
        db      06eh
        db      06eh
        db      06fh
        db      074h
        db      020h
        db      06fh
        db      070h
        db      065h
        db      06eh
        db      020h
        db      074h
        db      061h
        db      062h
        db      06ch
        db      065h
        db      020h
        db      00h

        code

        strings
L103:
        db      025h
        db      064h
        db      020h
        db      025h
        db      064h
        db      00h

        code

        strings
L104:
        db      025h
        db      064h
        db      00h

        code
```

```
        strings
L105:
        db      025h
        db      064h
        db      00h

        code

        strings
L110:
        db      020h
        db      0ah
        db      020h
        db      04dh
        db      041h
        db      04bh
        db      045h
        db      05fh
        db      04ch
        db      055h
        db      054h
        db      020h
        db      065h
        db      072h
        db      072h
        db      06fh
        db      072h
        db      020h
        db      020h
        db      020h
        db      020h
        db      025h
        db      073h
        db      020h
        db      00h

        code

        strings
```

```
L111:
        db      020h
        db      077h
        db      072h
        db      06fh
        db      06eh
        db      067h
        db      020h
        db      067h
        db      072h
        db      06fh
        db      075h
        db      070h
        db      069h
        db      06eh
        db      067h
        db      02dh
        db      020h
        db      06eh
        db      06fh
        db      074h
        db      020h
        db      033h
        db      020h
        db      06fh
        db      072h
        db      020h
        db      034h
        db      020h
        db      00h

        code

        strings
L113:
        db      020h
        db      0ah
        db      020h
        db      04dh
        db      041h
        db      04bh
        db      045h
        db      05fh
```

```
        db      04ch
        db      055h
        db      054h
        db      020h
        db      065h
        db      072h
        db      072h
        db      06fh
        db      072h
        db      020h
        db      020h
        db      020h
        db      020h
        db      025h
        db      073h
        db      020h
        db      00h
```

code

strings

```
L114:
        db      020h
        db      0ah
        db      020h
        db      063h
        db      061h
        db      06eh
        db      06eh
        db      06fh
        db      074h
        db      020h
        db      061h
        db      06ch
        db      06ch
        db      06fh
        db      063h
        db      061h
        db      074h
        db      065h
        db      020h
        db      073h
        db      070h
```

```
        db      061h
        db      063h
        db      065h
        db      020h
        db      066h
        db      06fh
        db      072h
        db      020h
        db      074h
        db      061h
        db      062h
        db      06ch
        db      065h
        db      020h
        db      00h

        code

        strings

L117:
        db      020h
        db      0ah
        db      020h
        db      04dh
        db      041h
        db      04bh
        db      045h
        db      05fh
        db      04ch
        db      055h
        db      054h
        db      020h
        db      065h
        db      072h
        db      072h
        db      06fh
        db      072h
        db      020h
        db      020h
        db      020h
        db      020h
        db      025h
        db      073h
```

```
          db      020h
          db      00h

          code

          strings

L118:
          db      0ah
          db      020h
          db      063h
          db      061h
          db      06eh
          db      06eh
          db      06fh
          db      074h
          db      020h
          db      072h
          db      065h
          db      061h
          db      064h
          db      020h
          db      074h
          db      061h
          db      062h
          db      06ch
          db      065h
          db      020h
          db      00h

          code

          strings

L121:
          db      020h
          db      0ah
          db      020h
          db      025h
          db      064h
          db      020h
          db      069h
          db      074h
```

```
        db      065h
        db      06dh
        db      073h
        db      020h
        db      072h
        db      065h
        db      061h
        db      064h
        db      020h
        db      00h

        code

        strings

L122:
        db      020h
        db      0ah
        db      020h
        db      04dh
        db      041h
        db      04bh
        db      045h
        db      05fh
        db      04ch
        db      055h
        db      054h
        db      020h
        db      065h
        db      072h
        db      072h
        db      06fh
        db      072h
        db      020h
        db      020h
        db      020h
        db      020h
        db      025h
        db      073h
        db      020h
        db      00h

        code
```

```
        strings

L123:
        db      0ah
        db      020h
        db      063h
        db      061h
        db      06eh
        db      06eh
        db      06fh
        db      074h
        db      020h
        db      072h
        db      065h
        db      061h
        db      064h
        db      020h
        db      074h
        db      061h
        db      062h
        db      06ch
        db      065h
        db      020h
        db      00h

        code


        strings

L125:
        db      020h
        db      045h
        db      06eh
        db      074h
        db      065h
        db      072h
        db      020h
        db      06eh
        db      061h
        db      06dh
        db      065h
        db      020h
        db      066h
```

```
        db      06fh
        db      072h
        db      020h
        db      06ch
        db      075h
        db      074h
        db      020h
        db      066h
        db      069h
        db      06ch
        db      065h
        db      020h
        db      00h

        code


        strings

L129:
        db      020h
        db      0ah
        db      020h
        db      04dh
        db      041h
        db      04bh
        db      045h
        db      05fh
        db      04ch
        db      055h
        db      054h
        db      020h
        db      065h
        db      072h
        db      072h
        db      06fh
        db      072h
        db      020h
        db      020h
        db      020h
        db      020h
        db      025h
        db      073h
        db      020h
        db      00h
```

```
        code

        strings

L130:
        db      0ah
        db      020h
        db      063h
        db      061h
        db      06eh
        db      06eh
        db      06fh
        db      074h
        db      020h
        db      061h
        db      06ch
        db      06ch
        db      06fh
        db      063h
        db      061h
        db      074h
        db      065h
        db      020h
        db      073h
        db      070h
        db      061h
        db      063h
        db      065h
        db      020h
        db      066h
        db      06fh
        db      072h
        db      020h
        db      077h
        db      065h
        db      069h
        db      067h
        db      068h
        db      074h
        db      05fh
        db      066h
        db      061h
        db      063h
```

```
            db      074h
            db      06fh
            db      072h
            db      020h
            db      00h

            code

            strings

L132:
            db      020h
            db      0ah
            db      020h
            db      04dh
            db      041h
            db      04bh
            db      045h
            db      05fh
            db      04ch
            db      055h
            db      054h
            db      020h
            db      065h
            db      072h
            db      072h
            db      06fh
            db      072h
            db      020h
            db      020h
            db      020h
            db      020h
            db      025h
            db      073h
            db      020h
            db      00h

            code

            strings

L133:
            db      0ah
```

```
        db      020h
        db      06fh
        db      076h
        db      065h
        db      072h
        db      066h
        db      06ch
        db      06fh
        db      077h
        db      020h
        db      069h
        db      06eh
        db      020h
        db      077h
        db      065h
        db      069h
        db      067h
        db      068h
        db      074h
        db      05fh
        db      066h
        db      061h
        db      063h
        db      074h
        db      06fh
        db      072h
        db      020h
        db      00h

        code

        strings

L136:
        db      020h
        db      0ah
        db      020h
        db      04dh
        db      041h
        db      04bh
        db      045h
        db      05fh
        db      04ch
        db      055h
```

```
        db      054h
        db      020h
        db      065h
        db      072h
        db      072h
        db      06fh
        db      072h
        db      020h
        db      020h
        db      020h
        db      020h
        db      025h
        db      073h
        db      020h
        db      00h

        code

        strings
L137:
        db      0ah
        db      020h
        db      063h
        db      061h
        db      06eh
        db      06eh
        db      06fh
        db      074h
        db      020h
        db      061h
        db      06ch
        db      06ch
        db      06fh
        db      063h
        db      061h
        db      074h
        db      065h
        db      020h
        db      073h
        db      070h
        db      061h
        db      063h
        db      065h
```

```
            db      020h
            db      066h
            db      06fh
            db      072h
            db      020h
            db      057h
            db      020h
            db      00h

            code

            strings

L139:
            db      020h
            db      0ah
            db      020h
            db      04dh
            db      041h
            db      04bh
            db      045h
            db      05fh
            db      04ch
            db      055h
            db      054h
            db      020h
            db      065h
            db      072h
            db      072h
            db      06fh
            db      072h
            db      020h
            db      020h
            db      020h
            db      020h
            db      025h
            db      073h
            db      020h
            db      00h

            code

            strings
```

```
L140:
        db      0ah
        db      020h
        db      06fh
        db      076h
        db      065h
        db      072h
        db      066h
        db      06ch
        db      06fh
        db      077h
        db      020h
        db      069h
        db      06eh
        db      020h
        db      057h
        db      020h
        db      00h

        code

        linkage

L10003:
        dw      03h

        code

        strings

L142:
        db      020h
        db      0ah
        db      020h
        db      04dh
        db      041h
        db      04bh
        db      045h
        db      05fh
        db      04ch
        db      055h
        db      054h
```

```
            db       020h
            db       065h
            db       072h
            db       072h
            db       06fh
            db       072h
            db       020h
            db       020h
            db       020h
            db       020h
            db       025h
            db       073h
            db       020h
            db       00h

            code

            strings

L143:
            db       0ah
            db       020h
            db       069h
            db       06eh
            db       063h
            db       06fh
            db       06dh
            db       070h
            db       061h
            db       074h
            db       069h
            db       062h
            db       06ch
            db       065h
            db       020h
            db       073h
            db       069h
            db       07ah
            db       065h
            db       073h
            db       020h
            db       00h

            code
```

```
        strings

L145:
        db      020h
        db      0ah
        db      020h
        db      04dh
        db      041h
        db      04bh
        db      045h
        db      05fh
        db      04ch
        db      055h
        db      054h
        db      020h
        db      065h
        db      072h
        db      072h
        db      06fh
        db      072h
        db      020h
        db      020h
        db      020h
        db      020h
        db      025h
        db      073h
        db      020h
        db      00h

        code

        strings

L146:
        db      0ah
        db      020h
        db      063h
        db      061h
        db      06eh
        db      06eh
        db      06fh
        db      074h
```

```
        db      020h
        db      061h
        db      06ch
        db      06ch
        db      06fh
        db      063h
        db      061h
        db      074h
        db      065h
        db      020h
        db      073h
        db      070h
        db      061h
        db      063h
        db      065h
        db      020h
        db      066h
        db      06fh
        db      072h
        db      020h
        db      077h
        db      065h
        db      069h
        db      067h
        db      068h
        db      074h
        db      05fh
        db      076h
        db      065h
        db      063h
        db      00h

        code

        strings

L148:
        db      020h
        db      0ah
        db      020h
        db      04dh
        db      041h
        db      04bh
        db      045h
```

```
        db      05fh
        db      04ch
        db      055h
        db      054h
        db      020h
        db      065h
        db      072h
        db      072h
        db      06fh
        db      072h
        db      020h
        db      020h
        db      020h
        db      020h
        db      025h
        db      073h
        db      020h
        db      00h

        code

        strings

L149:
        db      0ah
        db      020h
        db      063h
        db      061h
        db      06eh
        db      06eh
        db      06fh
        db      074h
        db      020h
        db      061h
        db      06ch
        db      06ch
        db      06fh
        db      063h
        db      061h
        db      074h
        db      065h
        db      020h
        db      073h
        db      070h
```

```
        db      061h
        db      063h
        db      065h
        db      020h
        db      066h
        db      06fh
        db      072h
        db      020h
        db      063h
        db      06ch
        db      075h
        db      074h
        db      020h
        db      00h

        code

        strings

L151:
        db      020h
        db      0ah
        db      020h
        db      04dh
        db      041h
        db      04bh
        db      045h
        db      05fh
        db      04ch
        db      055h
        db      054h
        db      020h
        db      065h
        db      072h
        db      072h
        db      06fh
        db      072h
        db      020h
        db      020h
        db      020h
        db      020h
        db      025h
        db      073h
        db      020h
```

```
                db      00h

                code

                strings

L152:
                db      0ah
                db      020h
                db      063h
                db      061h
                db      06eh
                db      06eh
                db      06fh
                db      074h
                db      020h
                db      061h
                db      06ch
                db      06ch
                db      06fh
                db      063h
                db      061h
                db      074h
                db      065h
                db      020h
                db      073h
                db      070h
                db      061h
                db      063h
                db      065h
                db      020h
                db      066h
                db      06fh
                db      072h
                db      020h
                db      06dh
                db      06ch
                db      075h
                db      074h
                db      00h

                code
```

```
        strings

L154:
        db      020h
        db      0ah
        db      020h
        db      04dh
        db      041h
        db      04bh
        db      045h
        db      05fh
        db      04ch
        db      055h
        db      054h
        db      020h
        db      065h
        db      072h
        db      072h
        db      06fh
        db      072h
        db      020h
        db      020h
        db      020h
        db      020h
        db      025h
        db      073h
        db      020h
        db      00h

        code

        strings

L155:
        db      0ah
        db      020h
        db      063h
        db      061h
        db      06eh
        db      06eh
        db      06fh
        db      074h
        db      020h
        db      061h
```

```
        db      06ch
        db      06ch
        db      06fh
        db      063h
        db      061h
        db      074h
        db      065h
        db      020h
        db      073h
        db      070h
        db      061h
        db      063h
        db      065h
        db      020h
        db      066h
        db      06fh
        db      072h
        db      020h
        db      079h
        db      06ch
        db      075h
        db      074h
        db      00h

        code

        strings

L157:
        db      020h
        db      0ah
        db      020h
        db      04dh
        db      041h
        db      04bh
        db      045h
        db      05fh
        db      04ch
        db      055h
        db      054h
        db      020h
        db      065h
        db      072h
        db      072h
```

```
            db      06fh
            db      072h
            db      020h
            db      020h
            db      020h
            db      020h
            db      025h
            db      073h
            db      020h
            db      00h

            code

            strings

        L158:
            db      0ah
            db      020h
            db      063h
            db      061h
            db      06eh
            db      06eh
            db      06fh
            db      074h
            db      020h
            db      061h
            db      06ch
            db      06ch
            db      06fh
            db      063h
            db      061h
            db      074h
            db      065h
            db      020h
            db      073h
            db      070h
            db      061h
            db      063h
            db      065h
            db      020h
            db      066h
            db      06fh
            db      072h
            db      020h
```

```
        db      06bh
        db      06ch
        db      075h
        db      074h
        db      00h

        code

        strings

L159:
        db      0ah
        db      020h
        db      043h
        db      06fh
        db      06dh
        db      070h
        db      075h
        db      074h
        db      065h
        db      020h
        db      074h
        db      068h
        db      065h
        db      020h
        db      06dh
        db      061h
        db      074h
        db      072h
        db      069h
        db      063h
        db      065h
        db      073h
        db      020h
        db      0ah
        db      00h

        code

        strings

L161:
        db      020h
```

```
          db      0ah
          db      020h
          db      04dh
          db      041h
          db      04bh
          db      045h
          db      05fh
          db      04ch
          db      055h
          db      054h
          db      020h
          db      065h
          db      072h
          db      072h
          db      06fh
          db      072h
          db      020h
          db      020h
          db      020h
          db      020h
          db      025h
          db      073h
          db      020h
          db      00h

          code

          strings
L162:
          db      0ah
          db      020h
          db      063h
          db      061h
          db      06eh
          db      06eh
          db      06fh
          db      074h
          db      020h
          db      061h
          db      06ch
          db      06ch
          db      06fh
          db      063h
```

```
        db      061h
        db      074h
        db      065h
        db      020h
        db      073h
        db      070h
        db      061h
        db      063h
        db      065h
        db      020h
        db      066h
        db      06fh
        db      072h
        db      020h
        db      04ch
        db      055h
        db      054h
        db      020h
        db      00h

        code

        strings
L163:
        db      0ah
        db      020h
        db      04ch
        db      045h
        db      056h
        db      045h
        db      04ch
        db      020h
        db      03dh
        db      020h
        db      031h
        db      020h
        db      020h
        db      02ah
        db      02ah
        db      02ah
        db      02ah
        db      02ah
        db      02ah
```

```
          db      02ah
          db      02ah
          db      02ah
          db      02ah
          db      02ah
          db      02ah
          db      02ah
          db      02ah
          db      02ah
          db      02ah
          db      02ah
          db      02ah
          db      02ah
          db      02ah
          db      02ah
          db      02ah
          db      02ah
          db      02ah
          db      02ah
          db      02ah
          db      020h
          db      0ah
          db      00h

          code

          strings

L174:
          db      0ah
          db      020h
          db      020h
          db      065h
          db      072h
          db      072h
          db      06fh
          db      072h
          db      020h
          db      069h
          db      06eh
          db      020h
          db      066h
          db      069h
          db      074h
```

```
        db      05fh
        db      06dh
        db      061h
        db      074h
        db      072h
        db      069h
        db      078h
        db      020h
        db      031h
        db      020h
        db      00h

        code

        strings

L176:
        db      0ah
        db      020h
        db      04ch
        db      045h
        db      056h
        db      045h
        db      04ch
        db      020h
        db      03dh
        db      020h
        db      032h
        db      020h
        db      020h
        db      02ah
        db      02ah
        db      02ah
        db      02ah
        db      02ah
        db      02ah
        db      02ah
        db      02ah
        db      02ah
        db      02ah
        db      02ah
        db      02ah
        db      02ah
        db      02ah
```

```
        db      02ah
        db      02ah
        db      02ah
        db      02ah
        db      02ah
        db      02ah
        db      02ah
        db      02ah
        db      02ah
        db      02ah
        db      02ah
        db      02ah
        db      020h
        db      0ah
        db      00h

        code

        strings

L180:
        db      020h
        db      025h
        db      064h
        db      00h

        code

        linkage

L10004:
        dw      020h

        code

        strings

L188:
        db      0ah
        db      020h
        db      065h
        db      072h
```

```
        db      072h
        db      06fh
        db      072h
        db      020h
        db      069h
        db      06eh
        db      020h
        db      066h
        db      069h
        db      074h
        db      05fh
        db      06dh
        db      061h
        db      074h
        db      072h
        db      069h
        db      078h
        db      020h
        db      032h
        db      020h
        db      00h

        code

        strings

L192:
        db      0ah
        db      020h
        db      043h
        db      06fh
        db      06dh
        db      070h
        db      075h
        db      074h
        db      065h
        db      020h
        db      074h
        db      068h
        db      065h
        db      020h
        db      06ch
        db      075h
        db      074h
```

```
        db      020h
        db      0ah
        db      00h

        code

        strings

L203:
        db      0ah
        db      020h
        db      072h
        db      067h
        db      062h
        db      03dh
        db      03dh
        db      020h
        db      025h
        db      064h
        db      020h
        db      025h
        db      064h
        db      020h
        db      025h
        db      064h
        db      0ah
        db      00h

        code

        strings

L204:
        db      020h
        db      0ah
        db      020h
        db      04dh
        db      041h
        db      04bh
        db      045h
        db      05fh
        db      04ch
        db      055h
```

```
        db      054h
        db      020h
        db      065h
        db      072h
        db      072h
        db      06fh
        db      072h
        db      020h
        db      020h
        db      020h
        db      020h
        db      025h
        db      073h
        db      020h
        db      00h

        code

        strings

L205:
        db      020h
        db      07ah
        db      065h
        db      072h
        db      06fh
        db      020h
        db      077h
        db      065h
        db      069h
        db      067h
        db      068h
        db      074h
        db      020h
        db      00h

        code

        linkage

L10005:
        db      00h
        db      00h
```

```
                db      00h
                db      00h
                db      00h
                db      00h
                db      0f0h
                db      03fh

                code

                linkage

L10006:
                db      00h
                db      00h
                db      00h
                db      00h
                db      00h
                db      00h
                db      0e0h
                db      03fh

                code

                linkage

L10007:
                dw      02h

                code

                strings

L216:
                db      020h
                db      057h
                db      072h
                db      069h
                db      074h
                db      065h
                db      020h
                db      074h
                db      068h
```

```
        db      065h
        db      020h
        db      06ch
        db      075h
        db      074h
        db      020h
        db      06fh
        db      06eh
        db      020h
        db      066h
        db      069h
        db      06ch
        db      065h
        db      03ah
        db      020h
        db      0ah
        db      020h
        db      00h

        code

        strings

L222:
        db      020h
        db      0ah
        db      020h
        db      04dh
        db      041h
        db      04bh
        db      045h
        db      05fh
        db      04ch
        db      055h
        db      054h
        db      020h
        db      065h
        db      072h
        db      072h
        db      06fh
        db      072h
        db      020h
        db      020h
        db      020h
```

```
        db      020h
        db      025h
        db      073h
        db      020h
        db      00h

        code

        push    si
        push    di
        enter   08ch, 00h
        mov     word ptr rgb_listlength, 00h
        sub     ax, ax
        mov     word ptr LUT, ax
        mov     word ptr rgb_list, ax
        mov     word ptr weight_factor, ax
        mov     word ptr W, ax
        sub     ax, ax
        mov     word ptr klut, ax
        mov     word ptr ylut, ax
        mov     word ptr mlut, ax
        mov     word ptr clut, ax
        mov     word ptr weight_vec, ax
        mov     word ptr A, ax
        sub     ax, ax
        mov     word ptr [bp-078h], ax
        mov     word ptr [bp-076h], ax
        mov     ax, word ptr size1
        sal     ax, 01h
        mov     word ptr msize, ax
        push    02h
        push    ax
        call    calloc
        add     sp, 04h
        mov     word ptr rgb_list, ax
        cmp     word ptr rgb_list, 00h
        jne     word ptr L98
        push    offset L96
        push    offset L95
L20082:
        call    printf
        add     sp, 04h
        mov     word ptr [bp-076h], -01h
        jmp     word ptr L97
L98:
```

```
push    word ptr CMYK_ptr
push    050h
lea     ax, word ptr [bp-062h]
push    ax
call    fgets
add     sp, 06h
lea     ax, word ptr [bp-070h]
push    ax
lea     ax, word ptr [bp-06eh]
push    ax
push    offset L103
push    word ptr CMYK_ptr
push    050h
lea     ax, word ptr [bp-062h]
push    ax
call    fgets
add     sp, 06h
push    ax
call    sscanf
add     sp, 08h
lea     ax, word ptr [bp-072h]
push    ax
push    offset L104
push    word ptr CMYK_ptr
push    050h
lea     ax, word ptr [bp-062h]
push    ax
call    fgets
add     sp, 06h
push    ax
call    sscanf
add     sp, 06h
lea     ax, word ptr [bp-074h]
push    ax
push    offset L105
push    word ptr CMYK_ptr
push    050h
lea     ax, word ptr [bp-062h]
push    ax
call    fgets
add     sp, 06h
push    ax
call    sscanf
add     sp, 06h
cmp     word ptr [bp-072h], 03h
```

```
         je      word ptr L20083
         jmp     word ptr L106
L20083:
         mov     word ptr k_flag, 00h
L107:
         mov     ax, word ptr [bp-074h]
         mov     word ptr msize, ax
         push    08h
         push    ax
         call    calloc
         add     sp, 04h
         mov     word ptr A, ax
         cmp     word ptr A, 00h
         jne     word ptr L112
         push    offset L114
         push    offset L113
         jmp     word ptr L20082
L112:
         push    00h
         push    00h
         push    03e8h
         push    word ptr CMYK_ptr
         call    fseek
         add     sp, 08h
         push    word ptr CMYK_ptr
         push    word ptr [bp-074h]
         push    08h
         push    word ptr A
         call    freadb
         add     sp, 08h
         mov     di, ax
         mov     ax, di
         cwd
         mov     word ptr sk, ax
         mov     word ptr sk+02h, dx
         or      ax, dx
         je      word ptr L124
         push    offset L118
         push    offset L117
         jmp     word ptr L20082
L124:
         mov     ax, word ptr [bp-074h]
         cwd
         idiv    word ptr [bp-072h]
         mov     word ptr msize, ax
```

```
            cmp     word ptr msize, 03e8h
            jae     word ptr L127
            mov     word ptr msize, 03e8h
L127:
            push    04h
            push    word ptr msize
            call    calloc
            add     sp, 04h
            mov     word ptr weight_factor, ax
            cmp     word ptr weight_factor, 00h
            jne     word ptr L128
            push    offset L130
            push    offset L129
            jmp     word ptr L20082
L128:
            cmp     word ptr msize, 07fffh
            jbe     word ptr L131
            push    offset L133
            push    offset L132
            jmp     word ptr L20082
L131:
            sal     word ptr msize, 02h
            cmp     word ptr msize, 03e8h
            jae     word ptr L134
            mov     word ptr msize, 03e8h
L134:
            push    04h
            push    word ptr msize
            call    calloc
            add     sp, 04h
            mov     word ptr W, ax
            cmp     word ptr W, 00h
            jne     word ptr L135
            push    offset L137
            push    offset L136
            jmp     word ptr L20082
L135:
            cmp     word ptr msize, 03fffh
            jbe     word ptr L138
            push    offset L140
            push    offset L139
            jmp     word ptr L20082
L138:
            mov     ax, word ptr [bp-074h]
            cwd
```

```
        idiv    word ptr [bp-072h]
        mov     di, ax
        mov     ax, word ptr size1
        cwd
        idiv    cs:word ptr L10003
        cmp     ax, di
        je      word ptr L141
        push    offset L143
        push    offset L142
        jmp     word ptr L20082
L141:
        mov     word ptr msize, 01331h
        push    08h
        push    word ptr msize
        call    calloc
        add     sp, 04h
        mov     word ptr weight_vec, ax
        cmp     word ptr weight_vec, 00h
        jne     word ptr L144
        push    offset L146
        push    offset L145
        jmp     word ptr L20082
L144:
        push    08h
        push    word ptr msize
        call    calloc
        add     sp, 04h
        mov     word ptr clut, ax
        cmp     word ptr clut, 00h
        jne     word ptr L147
        push    offset L149
        push    offset L148
        jmp     word ptr L20082
L147:
        push    08h
        push    word ptr msize
        call    calloc
        add     sp, 04h
        mov     word ptr mlut, ax
        cmp     word ptr mlut, 00h
        jne     word ptr L150
        push    offset L152
        push    offset L151
        jmp     word ptr L20082
L150:
```

```
        push    08h
        push    word ptr msize
        call    calloc
        add     sp, 04h
        mov     word ptr ylut, ax
        cmp     word ptr ylut, 00h
        jne     word ptr L153
        push    offset L155
        push    offset L154
        jmp     word ptr L20082
L153:
        push    08h
        push    word ptr msize
        call    calloc
        add     sp, 04h
        mov     word ptr klut, ax
        cmp     word ptr klut, 00h
        jne     word ptr L156
        push    offset L158
        push    offset L157
        jmp     word ptr L20082
L156:
        push    offset L159
        call    printf
        add     sp, 02h
        mov     word ptr msize, 01331h
        push    08h
        push    word ptr msize
        call    calloc
        add     sp, 04h
        mov     word ptr LUT, ax
        cmp     word ptr LUT, 00h
        jne     word ptr L160
        push    offset L162
        push    offset L161
        jmp     word ptr L20082
L160:
        push    offset L163
        call    printf
        add     sp, 02h
        mov     word ptr [bp-0ah], 040h
L166:
        cmp     word ptr [bp-0ah], 0100h
        jg      word ptr L164
        mov     word ptr [bp-08h], 040h
```

```
L169:
        cmp     word ptr [bp-08h], 0100h
        jg      word ptr L167
        mov     word ptr [bp-06h], 040h
L172:
        cmp     word ptr [bp-06h], 0100h
        jg      word ptr L170
        push    01h
        lea     ax, word ptr [bp-0ah]
        push    ax
        call    fit_matrix
        add     sp, 04h
        mov     word ptr [bp-076h], ax
        cmp     word ptr [bp-076h], 00h
        jge     word ptr L173
        push    offset L174
        call    printf
        add     sp, 02h
L173:
        cmp     word ptr [bp-076h], 00h
        jne     word ptr L175
        push    01h
        lea     ax, word ptr [bp-0ah]
        push    ax
        call    getlut
        add     sp, 04h
L175:
        add     word ptr [bp-06h], 080h
        jmp     word ptr L172
L170:
        add     word ptr [bp-08h], 080h
        jmp     word ptr L169
L167:
        add     word ptr [bp-0ah], 080h
        jmp     word ptr L166
L164:
        push    offset L176
        call    printf
        add     sp, 02h
        mov     word ptr [bp-0ah], 00h
L179:
        cmp     word ptr [bp-0ah], 0100h
        jle     word ptr L20084
        jmp     word ptr L177
L20084:
```

```
        mov     ax, 0100h
        sub     ax, word ptr [bp-0ah]
        cwd
        idiv    cs:word ptr L10004
        inc     ax
        push    ax
        push    offset L180
        call    printf
        add     sp, 04h
        push    offset _file+0eh
        call    fflush
        add     sp, 02h
        mov     word ptr [bp-08h], 00h
L183:
        cmp     word ptr [bp-08h], 0100h
        jg      word ptr L181
        mov     word ptr [bp-06h], 00h
L186:
        cmp     word ptr [bp-06h], 0100h
        jg      word ptr L184
        push    02h
        lea     ax, word ptr [bp-0ah]
        push    ax
        call    fit_matrix
        add     sp, 04h
        mov     word ptr [bp-076h], ax
        cmp     word ptr [bp-076h], 00h
        jge     word ptr L187
        push    offset L188
        call    printf
        add     sp, 02h
L187:
        cmp     word ptr [bp-076h], 00h
        jne     word ptr L189
        push    02h
        jmp     word ptr L20029
L189:
        cmp     word ptr [bp-076h], 01h
        jne     word ptr L190
        push    03h
L20029:
        lea     ax, word ptr [bp-0ah]
        push    ax
        call    getlut
        add     sp, 04h
```

```
L190:
        add     word ptr [bp-06h], 020h
        jmp     word ptr L186
L184:
        add     word ptr [bp-08h], 020h
        jmp     word ptr L183
L181:
        add     word ptr [bp-0ah], 020h
        jmp     word ptr L179
L177:
        mov     word ptr [bp-076h], 00h
        push    offset L192
        call    printf
        add     sp, 02h
        lea     ax, word ptr [bp-012h]
        mov     word ptr [bp-07ah], ax
        mov     word ptr [bp-066h], 00h
L195:
        cmp     word ptr [bp-066h], 011h
        jl      word ptr L20085
        jmp     word ptr L193
L20085:
        mov     word ptr [bp-068h], 00h
L198:
        cmp     word ptr [bp-068h], 011h
        jl      word ptr L20086
        jmp     word ptr L196
L20086:
        mov     word ptr [bp-06ah], 00h
L201:
        cmp     word ptr [bp-06ah], 011h
        jl      word ptr L20087
        jmp     word ptr L199
L20087:
        imul    ax, word ptr [bp-068h], 011h
        mov     di, ax
        imul    ax, word ptr [bp-06ah], 0121h
        add     ax, di
        add     ax, word ptr [bp-066h]
        mov     word ptr [bp-07ch], ax
        mov     di, word ptr [bp-07ch]
        sal     di, 03h
        add     di, word ptr weight_vec
        fldd    word ptr [di]
        fstpd   word ptr [bp-08ch]
```

```
        fldd    word ptr [bp-08ch]
        fcompd  word ptr epsilon
        call    cfcc
        jae     word ptr L202
        mov     ax, word ptr [bp-06ah]
        sal     ax, 04h
        push    ax
        mov     ax, word ptr [bp-068h]
        sal     ax, 04h
        push    ax
        mov     ax, word ptr [bp-066h]
        sal     ax, 04h
        push    ax
        push    offset L203
        call    printf
        add     sp, 08h
        push    offset L205
        push    offset L204
        jmp     word ptr L20082
L202:
        fldd    cs:word ptr L10005
        fdivd   word ptr [bp-08ch]
        fstpd   word ptr [bp-08ch]
        mov     di, word ptr [bp-07ch]
        sal     di, 03h
        add     di, word ptr clut
        fldd    word ptr [di]
        fmuld   word ptr [bp-08ch]
        fstpd   word ptr [bp-084h]
        fldd    word ptr [bp-084h]
        faddd   cs:word ptr L10006
        call    dp87
        call    idcvt
        add     sp, 08h
        mov     di, word ptr [bp-07ah]
        mov     word ptr [di], ax
        mov     di, word ptr [bp-07ch]
        sal     di, 03h
        add     di, word ptr mlut
        fldd    word ptr [di]
        fmuld   word ptr [bp-08ch]
        fstpd   word ptr [bp-084h]
        fldd    word ptr [bp-084h]
        faddd   cs:word ptr L10006
        call    dp87
```

```
call    idcvt
add     sp, 08h
mov     di, word ptr [bp-07ah]
mov     word ptr [di+02h], ax
mov     di, word ptr [bp-07ch]
sal     di, 03h
add     di, word ptr ylut
fldd    word ptr [di]
fmuld   word ptr [bp-08ch]
fstpd   word ptr [bp-084h]
fldd    word ptr [bp-084h]
faddd   cs:word ptr L10006
call    dp87
call    idcvt
add     sp, 08h
mov     di, word ptr [bp-07ah]
mov     word ptr [di+04h], ax
mov     di, word ptr [bp-07ch]
sal     di, 03h
add     di, word ptr klut
fldd    word ptr [di]
fmuld   word ptr [bp-08ch]
fstpd   word ptr [bp-084h]
fldd    word ptr [bp-084h]
faddd   cs:word ptr L10006
call    dp87
call    idcvt
add     sp, 08h
mov     di, word ptr [bp-07ah]
mov     word ptr [di+06h], ax
push    08h
lea     ax, word ptr [bp-012h]
push    ax
imul    ax, word ptr [bp-068h], 011h
mov     di, ax
imul    ax, word ptr [bp-06ah], 0121h
add     ax, di
add     ax, word ptr [bp-066h]
sal     ax, 03h
add     ax, word ptr LUT
push    ax
call    blkmv
add     sp, 06h
inc     word ptr [bp-06ah]
jmp     word ptr L201
```

```
L199:
        inc     word ptr [bp-068h]
        jmp     word ptr L198
L196:
        inc     word ptr [bp-066h]
        jmp     word ptr L195
L193:
        cmp     word ptr shadow_contrast_flag, 00h
        jne     word ptr L20088
        jmp     word ptr L206
L20088:
        mov     ax, word ptr endc
        mov     di, word ptr [bp-07ah]
        mov     word ptr [di], ax
        mov     ax, word ptr endm
        mov     word ptr [di+02h], ax
        mov     ax, word ptr endy
        mov     word ptr [di+04h], ax
        mov     ax, word ptr endk
        mov     word ptr [di+06h], ax
        push    08h
        lea     ax, word ptr [bp-012h]
        push    ax
        push    word ptr LUT
        call    blkmv
        add     sp, 06h
        mov     ax, word ptr endc
        cwd
        idiv    cs:word ptr L10007
        mov     word ptr endc, ax
        mov     ax, word ptr endm
        cwd
        idiv    cs:word ptr L10007
        mov     word ptr endm, ax
        mov     ax, word ptr endy
        cwd
        idiv    cs:word ptr L10007
        mov     word ptr endy, ax
        mov     ax, word ptr endk
        cwd
        idiv    cs:word ptr L10007
        mov     word ptr endk, ax
        lea     ax, word ptr [bp-012h]
        mov     word ptr [bp-07ah], ax
        mov     word ptr [bp-066h], 01h
```

```
L209:
        cmp     word ptr [bp-066h], 00h
        jge     word ptr L20089
        jmp     word ptr L206
L20089:
        mov     word ptr [bp-068h], 01h
L212:
        cmp     word ptr [bp-068h], 00h
        jge     word ptr L20090
        jmp     word ptr L210
L20090:
        mov     word ptr [bp-06ah], 01h
L215:
        cmp     word ptr [bp-06ah], 00h
        jge     word ptr L20091
        jmp     word ptr L213
L20091:
        push    08h
        mov     di, word ptr [bp-068h]
        add     di, di
        imul    ax, di, 011h
        mov     si, ax
        mov     di, word ptr [bp-06ah]
        add     di, di
        imul    ax, di, 0121h
        add     ax, si
        add     ax, word ptr [bp-066h]
        add     ax, word ptr [bp-066h]
        sal     ax, 03h
        add     ax, word ptr LUT
        push    ax
        lea     ax, word ptr [bp-012h]
        push    ax
        call    blkmv
        add     sp, 06h
        mov     di, word ptr [bp-07ah]
        mov     ax, word ptr [di]
        cwd
        idiv    cs:word ptr L10007
        mov     word ptr [di], ax
        mov     ax, word ptr endc
        add     word ptr [di], ax
        mov     ax, word ptr [di+02h]
        cwd
        idiv    cs:word ptr L10007
```

```
        mov     word ptr [di+02h], ax
        mov     ax, word ptr endm
        add     word ptr [di+02h], ax
        mov     ax, word ptr [di+04h]
        cwd
        idiv    cs:word ptr L10007
        mov     word ptr [di+04h], ax
        mov     ax, word ptr endy
        add     word ptr [di+04h], ax
        mov     ax, word ptr [di+06h]
        cwd
        idiv    cs:word ptr L10007
        mov     word ptr [di+06h], ax
        mov     ax, word ptr endk
        add     word ptr [di+06h], ax
        push    08h
        lea     ax, word ptr [bp-012h]
        push    ax
        imul    ax, word ptr [bp-068h], 011h
        mov     di, ax
        imul    ax, word ptr [bp-06ah], 0121h
        add     ax, di
        add     ax, word ptr [bp-066h]
        sal     ax, 03h
        add     ax, word ptr LUT
        push    ax
        call    blkmv
        add     sp, 06h
        dec     word ptr [bp-06ah]
        jmp     word ptr L215
L213:
        dec     word ptr [bp-068h]
        jmp     word ptr L212
L210:
        dec     word ptr [bp-066h]
        jmp     word ptr L209
L206:
        push    offset L216
        call    printf
        add     sp, 02h
        jmp     word ptr L217
L220:
        cmp     word ptr [bp-064h], 04h
        jge     word ptr L217
        push    word ptr clut
```

```
            mov     ax, word ptr [bp-064h]
            sal     ax, 01h
            add     ax, word ptr [bp-07ah]
            push    ax
            call    arrange_lut
            add     sp, 04h
            push    word ptr [bp-04h]
            push    02000h
            push    02h
            push    word ptr clut
            call    fwrite
            add     sp, 08h
            mov     word ptr [bp-06ch], ax
            cmp     ax, 02000h
            je      word ptr L221
            push    0ah
            push    offset L222
            jmp     word ptr L20082
    L221:
            inc     word ptr [bp-064h]
            jmp     word ptr L220
    L217:
            call    get_buf3d
            mov     ax, word ptr LUT
            mov     word ptr [bp-07ah], ax
            mov     word ptr [bp-064h], 00h
    L225:
            cmp     word ptr [bp-064h], 04h
            jge     word ptr L223
            mov     ax, word ptr [bp-064h]
            sal     ax, 0eh
            add     ax, word ptr buf_current
            push    ax
            mov     ax, word ptr [bp-064h]
            sal     ax, 01h
            add     ax, word ptr [bp-07ah]
            push    ax
            call    arrange_lut
            add     sp, 04h
            inc     word ptr [bp-064h]
            jmp     word ptr L225
    L223:
            call    init_grad
            mov     word ptr gcor_lut, 01h
            push    word ptr buf_current
```

```
        call    load3d
        add     sp, 02h
        mov     word ptr LUT_TYPE, 0ah
        mov     ax, word ptr D_12
        mov     word ptr DARK12, ax
        mov     ax, word ptr W_12
        mov     word ptr WHITE12, ax
        push    00h
        push    00h
        push    00h
        push    02719h
        call    handle_exc
        add     sp, 08h
        jmp     word ptr L97
L106:
        cmp     word ptr [bp-072h], 04h
        jne     word ptr L108
        mov     word ptr k_flag, 01h
        jmp     word ptr L107
L108:
        push    offset L111
        push    offset L110
        jmp     word ptr L20082
L97:
        cmp     word ptr [bp-076h], 00h
        jge     word ptr L226
        push    00h
        push    00h
        push    00h
        push    0271ch
        call    handle_exc
        add     sp, 08h
L226:
        cmp     word ptr W, 00h
        je      word ptr L227
        push    word ptr W
        call    free
        add     sp, 02h
L227:
        cmp     word ptr weight_factor, 00h
        je      word ptr L228
        push    word ptr weight_factor
        call    free
        add     sp, 02h
L228:
```

```
          cmp     word ptr rgb_list, 00h
          je      word ptr L229
          push    word ptr rgb_list
          call    free
          add     sp, 02h
L229:
          cmp     word ptr LUT, 00h
          je      word ptr L230
          push    word ptr LUT
          call    free
          add     sp, 02h
L230:
          cmp     word ptr A, 00h
          je      word ptr L231
          push    word ptr A
          call    free
          add     sp, 02h
L231:
          cmp     word ptr weight_vec, 00h
          je      word ptr L232
          push    word ptr weight_vec
          call    free
          add     sp, 02h
L232:
          cmp     word ptr clut, 00h
          je      word ptr L233
          push    word ptr clut
          call    free
          add     sp, 02h
L233:
          cmp     word ptr mlut, 00h
          je      word ptr L234
          push    word ptr mlut
          call    free
          add     sp, 02h
L234:
          cmp     word ptr ylut, 00h
          je      word ptr L235
          push    word ptr ylut
          call    free
          add     sp, 02h
L235:
          cmp     word ptr klut, 00h
          je      word ptr L236
          push    word ptr klut
```

```
            call     free
            add      sp, 02h
L236:
            cmp      word ptr list, 00h
            je       word ptr L237
            push     word ptr list
            call     free
            add      sp, 02h
L237:
            cmp      word ptr root, 00h
            je       word ptr L238
            push     word ptr root
            call     free
            add      sp, 02h
L238:
            push     word ptr CMYK_ptr
            call     fclose
            add      sp, 02h
            mov      ax, word ptr [bp-078h]
            fwait
            leave
            pop      di
            pop      si
            ret
            public fit_matrix
fit_matrix:

            strings

L245:
            db       0ah
            db       020h
            db       025h
            db       064h
            db       020h
            db       065h
            db       06ch
            db       065h
            db       06dh
            db       065h
            db       06eh
            db       074h
            db       073h
            db       020h
            db       02dh
```

```
        db      020h
        db      074h
        db      06fh
        db      06fh
        db      020h
        db      06dh
        db      061h
        db      06eh
        db      079h
        db      020h
        db      00h

        code

        strings

L265:
        db      0ah
        db      020h
        db      025h
        db      064h
        db      020h
        db      065h
        db      06ch
        db      065h
        db      06dh
        db      065h
        db      06eh
        db      074h
        db      073h
        db      020h
        db      02dh
        db      020h
        db      074h
        db      06fh
        db      06fh
        db      020h
        db      06dh
        db      061h
        db      06eh
        db      079h
        db      020h
        db      00h
```

```
        code

        strings

L271:
        db      0ah
        db      020h
        db      06eh
        db      06fh
        db      020h
        db      073h
        db      06fh
        db      06ch
        db      075h
        db      074h
        db      069h
        db      06fh
        db      06eh
        db      020h
        db      066h
        db      06fh
        db      072h
        db      020h
        db      072h
        db      067h
        db      062h
        db      020h
        db      03dh
        db      020h
        db      025h
        db      064h
        db      020h
        db      025h
        db      064h
        db      020h
        db      025h
        db      064h
        db      020h
        db      02ch
        db      020h
        db      020h
        db      025h
        db      064h
        db      020h
```

```
        db      065h
        db      06ch
        db      065h
        db      06dh
        db      065h
        db      06eh
        db      074h
        db      073h
        db      020h
        db      0ah
        db      020h
        db      00h

        code

        push    si
        push    di
        enter   028h, 00h
        movb    byte ptr [bp-01h], 061h
        movb    byte ptr [bp-03h], 041h
        mov     ax, 0100h
        mov     cx, word ptr [bp+0ah]
        sar     ax, cl
        mov     word ptr [bp-08h], ax
        sar     ax, 01h
        mov     word ptr [bp-0ah], ax
        mov     di, word ptr [bp+08h]
        and     ax, word ptr [di]
        je      word ptr L240
        mov     ax, word ptr [bp-0ah]
        and     ax, word ptr [di+02h]
        je      word ptr L240
        mov     ax, word ptr [bp-0ah]
        and     ax, word ptr [di+04h]
        je      word ptr L240
        mov     word ptr [bp-06h], 00h
        jmp     word ptr L241
L240:
        mov     word ptr [bp-06h], 01h
L241:
        cmp     word ptr [bp-06h], 00h
        jne     word ptr L242
        push    word ptr [bp+0ah]
        push    word ptr [bp+08h]
        call    findleaf
```

```
            add     sp, 04h
            mov     word ptr [bp-016h], ax
            cmp     word ptr [bp-016h], 00h
            jg      word ptr L243
            cmp     word ptr [bp+0ah], 00h
            je      word ptr L243
L20092:
            mov     ax, 02h
            jmp     word ptr L239
L243:
            imul    ax, word ptr [bp-016h], 014h
            mov     di, ax
            add     di, word ptr root
            mov     ax, word ptr [di]
            mov     word ptr [bp-0ch], ax
            cmp     word ptr [bp-0ch], 01e0h
            jl      word ptr L244
            push    ax
            push    offset L245
L20095:
            call    printf
            add     sp, 04h
L20096:
            mov     ax, -01h
            jmp     word ptr L239
L244:
            mov     word ptr rgb_listlength, 00h
            imul    ax, word ptr [bp-016h], 014h
            add     ax, word ptr root
            push    ax
            call    readtree
            add     sp, 02h
L246:
            mov     ax, word ptr [bp-0ch]
            cmp     ax, word ptr min_few
            jle     word ptr L20092
            mov     di, word ptr [bp+0ah]
            sal     di, 01h
            cmp     ax, word ptr [di+minval]
            jl      word ptr L20097
            jmp     word ptr L268
L20097:
            call    getfew
L269:
            sub     ax, ax
```

```
            jmp     word ptr L239
L242:
            cmp     word ptr [bp-06h], 01h
            jne     word ptr L20096
            inc     word ptr [bp+0ah]
            mov     word ptr rgb_listlength, 00h
            mov     word ptr [bp-0ch], 00h
            mov     word ptr [bp-024h], -05h
L250:
            cmp     word ptr [bp-024h], 0ah
            jl      word ptr L20098
            jmp     word ptr L248
L20098:
            mov     di, word ptr [bp+08h]
            mov     ax, word ptr [bp-024h]
            add     ax, word ptr [di]
            mov     word ptr [bp-012h], ax
            cmp     word ptr [bp-012h], 00h
            jg      word ptr L20099
            jmp     word ptr L251
L20099:
            cmp     word ptr [bp-012h], 0100h
            jl      word ptr L20100
            jmp     word ptr L251
L20100:
            mov     word ptr [bp-026h], -05h
L254:
            cmp     word ptr [bp-026h], 0ah
            jl      word ptr L20101
            jmp     word ptr L251
L20101:
            mov     di, word ptr [bp+08h]
            mov     ax, word ptr [bp-026h]
            add     ax, word ptr [di+02h]
            mov     word ptr [bp-010h], ax
            cmp     word ptr [bp-010h], 00h
            jg      word ptr L20102
            jmp     word ptr L255
L20102:
            cmp     word ptr [bp-010h], 0100h
            jl      word ptr L20103
            jmp     word ptr L255
L20103:
            mov     word ptr [bp-028h], -05h
L258:
```

```
          cmp     word ptr [bp-028h], 0ah
          jl      word ptr L20104
          jmp     word ptr L255
L20104:
          mov     di, word ptr [bp+08h]
          mov     ax, word ptr [bp-028h]
          add     ax, word ptr [di+04h]
          mov     word ptr [bp-0eh], ax
          cmp     word ptr [bp-0eh], 00h
          jle     word ptr L259
          cmp     word ptr [bp-0eh], 0100h
          jge     word ptr L259
          push    word ptr [bp+0ah]
          lea     ax, word ptr [bp-012h]
          push    ax
          call    findleaf
          add     sp, 04h
          mov     word ptr [bp-016h], ax
          cmp     word ptr [bp-016h], 00h
          je      word ptr L259
          cmp     word ptr [bp-016h], 00h
          jge     word ptr L261
          neg     ax
          push    ax
          push    word ptr [bp+0ah]
          lea     ax, word ptr [bp-012h]
          push    ax
          call    checkval
          add     sp, 06h
          mov     word ptr [bp-014h], ax
          cmp     word ptr [bp-014h], 01h
          jne     word ptr L259
          mov     di, word ptr rgb_listlength
          inc     word ptr rgb_listlength
          sal     di, 01h
          add     di, word ptr rgb_list
          mov     ax, word ptr [bp-016h]
          neg     ax
          mov     word ptr [di], ax
          inc     word ptr [bp-0ch]
          jmp     word ptr L259
L261:
          imul    ax, word ptr [bp-016h], 014h
          mov     di, ax
          add     di, word ptr root
```

```
            mov     ax, word ptr [di]
            add     word ptr [bp-0ch], ax
            imul    ax, word ptr [bp-016h], 014h
            add     ax, word ptr root
            push    ax
            call    readtree
            add     sp, 02h
L259:
            add     word ptr [bp-028h], 0ah
            jmp     word ptr L258
L255:
            add     word ptr [bp-026h], 0ah
            jmp     word ptr L254
L251:
            add     word ptr [bp-024h], 0ah
            jmp     word ptr L250
L248:
            dec     word ptr [bp+0ah]
            cmp     word ptr [bp-0ch], 01e0h
            jge     word ptr L20105
            jmp     word ptr L246
L20105:
            push    word ptr [bp-0ch]
            push    offset L265
            jmp     word ptr L20095
L268:
            push    word ptr [bp-08h]
            push    word ptr [bp+08h]
            call    getR
            add     sp, 04h
            call    getmat
            mov     word ptr [bp-014h], ax
            cmp     word ptr [bp-014h], 00h
            jl      word ptr L20106
            jmp     word ptr L269
L20106:
            push    word ptr [bp-0ch]
            mov     di, word ptr [bp+08h]
            push    word ptr [di+04h]
            push    word ptr [di+02h]
            push    word ptr [di]
            push    offset L271
            call    printf
            add     sp, 0ah
            mov     ax, word ptr [bp-014h]
```

```
L239:
        leave
        pop     di
        pop     si
        ret
        public getR
getR:

        linkage

L10008:
        dw      0100h

        code


        linkage

L10009:
        db      00h
        db      00h
        db      00h
        db      00h
        db      00h
        db      088h
        db      0c3h
        db      040h

        code


        linkage

L10010:
        db      00h
        db      00h
        db      00h
        db      00h
        db      00h
        db      06ah
        db      0f8h
        db      040h

        code
```

```
            linkage
    L10011:
            dw       0ah

            code

            push     si
            push     di
            enter    084h, 00h
            mov      word ptr [bp-04h], 00h
            mov      word ptr [bp-06h], 00h
    L275:
            cmp      word ptr [bp-06h], 03h
            jge      word ptr L273
            mov      di, word ptr [bp-06h]
            sal      di, 03h
            add      di, bp
            fldz
            fstpd    word ptr [di-05ch]
            inc      word ptr [bp-06h]
            jmp      word ptr L275
    L273:
            mov      word ptr [bp-06h], 00h
    L278:
            cmp      word ptr [bp-06h], 04h
            jge      word ptr L276
            mov      di, word ptr [bp-06h]
            sal      di, 03h
            add      di, bp
            fldz
            fstpd    word ptr [di-07ch]
            inc      word ptr [bp-06h]
            jmp      word ptr L278
    L276:
            mov      word ptr [bp-0ch], 03h
            cmp      word ptr k_flag, 00h
            je       word ptr L279
            mov      word ptr [bp-0ch], 04h
    L279:
            fldd     cs:word ptr L10005
            fdivi    word ptr [bp+0ah]
            fstpd    word ptr [bp-034h]
            mov      word ptr [bp-06h], 00h
```

```
L282:
        mov     ax, word ptr [bp-06h]
        cmp     ax, word ptr rgb_listlength
        jl      word ptr L20108
        jmp     word ptr L280
L20108:
        mov     di, word ptr [bp-06h]
        sal     di, 01h
        add     di, word ptr rgb_list
        mov     ax, word ptr [di]
        mov     word ptr [bp-02h], ax
        mov     di, word ptr [bp-04h]
        sal     di, 02h
        add     di, word ptr W
        fldd    cs:word ptr L10005
        fstpf   word ptr [di]
        mov     di, word ptr [bp-02h]
        sal     di, 03h
        add     di, word ptr list
        fldd    word ptr [di]
        mov     di, word ptr [bp-04h]
        inc     di
        sal     di, 02h
        add     di, word ptr W
        fstf    word ptr [di]
        fstpd   word ptr [bp-014h]
        mov     di, word ptr [bp-02h]
        inc     di
        sal     di, 03h
        add     di, word ptr list
        fldd    word ptr [di]
        mov     di, word ptr [bp-04h]
        inc     di
        inc     di
        sal     di, 02h
        add     di, word ptr W
        fstf    word ptr [di]
        fstpd   word ptr [bp-01ch]
        mov     di, word ptr [bp-02h]
        inc     di
        inc     di
        sal     di, 03h
        add     di, word ptr list
        fldd    word ptr [di]
        mov     di, word ptr [bp-04h]
```

```
              add     di, 03h
              sal     di, 02h
              add     di, word ptr W
              fstf    word ptr [di]
              fstpd   word ptr [bp-024h]
              fldd    word ptr [bp-014h]
              faddd   word ptr [bp-05ch]
              fstpd   word ptr [bp-05ch]
              fldd    word ptr [bp-01ch]
              faddd   word ptr [bp-054h]
              fstpd   word ptr [bp-054h]
              fldd    word ptr [bp-024h]
              faddd   word ptr [bp-04ch]
              fstpd   word ptr [bp-04ch]
              dec     ax
              imul    word ptr [bp-0ch]
              cwd
              idiv    cs:word ptr L10003
              mov     word ptr [bp-0ah], ax
              mov     word ptr [bp-08h], 00h
     L285:
              cmp     word ptr [bp-08h], 03h
              jge     word ptr L283
              mov     di, word ptr [bp-08h]
              add     di, word ptr [bp-0ah]
              sal     di, 03h
              add     di, word ptr A
              fldd    word ptr [di]
              mov     si, word ptr [bp-08h]
              sal     si, 02h
              mov     di, word ptr [bp-06h]
              sal     di, 04h
              add     di, si
              fstpf   word ptr [di+cmyk_list]
              inc     word ptr [bp-08h]
              jmp     word ptr L285
     L283:
              cmp     word ptr k_flag, 00h
              je      word ptr L286
              mov     di, word ptr [bp-0ah]
              add     di, 03h
              sal     di, 03h
              add     di, word ptr A
              fldd    word ptr [di]
              mov     di, word ptr [bp-06h]
```

```
                sal     di, 04h
                jmp     word ptr L20107
        L286:
                mov     di, word ptr [bp-06h]
                sal     di, 04h
                fldi    cs:word ptr L10008
        L20107:
                fstpf   word ptr [di+cmyk_list+0ch]
                mov     word ptr [bp-08h], 00h
        L290:
                cmp     word ptr [bp-08h], 04h
                jge     word ptr L288
                mov     si, word ptr [bp-08h]
                sal     si, 02h
                mov     di, word ptr [bp-06h]
                sal     di, 04h
                add     di, si
                fldf    word ptr [di+cmyk_list]
                mov     di, word ptr [bp-08h]
                sal     di, 03h
                add     di, bp
                faddd   word ptr [di-07ch]
                fstpd   word ptr [di-07ch]
                inc     word ptr [bp-08h]
                jmp     word ptr L290
        L288:
                mov     si, word ptr [bp+08h]
                mov     di, word ptr [bp-02h]
                sal     di, 03h
                add     di, word ptr list
                fldd    word ptr [di]
                fsubi   word ptr [si]
                fstpd   word ptr [bp-03ch]
                fldz
                fcompd  word ptr [bp-03ch]
                call    cfcc
                jbe     word ptr L291
                fldd    word ptr [bp-03ch]
                fchs
                fstpd   word ptr [bp-03ch]
        L291:
                mov     si, word ptr [bp+08h]
                mov     di, word ptr [bp-02h]
                inc     di
                sal     di, 03h
```

```
        add     di, word ptr list
        fldd    word ptr [di]
        fsubi   word ptr [si+02h]
        fstpd   word ptr [bp-044h]
        fldz
        fcompd  word ptr [bp-044h]
        call    cfcc
        jbe     word ptr L292
        fldd    word ptr [bp-044h]
        fchs
        fstpd   word ptr [bp-044h]
L292:
        fldd    word ptr [bp-03ch]
        fcompd  word ptr [bp-044h]
        call    cfcc
        jae     word ptr L293
        fldd    word ptr [bp-044h]
        fstpd   word ptr [bp-03ch]
L293:
        mov     si, word ptr [bp+08h]
        mov     di, word ptr [bp-02h]
        inc     di
        inc     di
        sal     di, 03h
        add     di, word ptr list
        fldd    word ptr [di]
        fsubi   word ptr [si+04h]
        fstpd   word ptr [bp-044h]
        fldz
        fcompd  word ptr [bp-044h]
        call    cfcc
        jbe     word ptr L294
        fldd    word ptr [bp-044h]
        fchs
        fstpd   word ptr [bp-044h]
L294:
        fldd    word ptr [bp-03ch]
        fcompd  word ptr [bp-044h]
        call    cfcc
        jae     word ptr L295
        fldd    word ptr [bp-044h]
        fstpd   word ptr [bp-03ch]
L295:
        fldi    cs:word ptr L10003
        fstpd   word ptr [bp-084h]
```

```
        fldd    word ptr [bp-014h]
        faddd   word ptr [bp-01ch]
        faddd   word ptr [bp-024h]
        fdivd   word ptr [bp-084h]
        fstpd   word ptr [bp-02ch]
        fldd    word ptr [bp-02ch]
        frsubd  word ptr [bp-014h]
        fstpd   word ptr [bp-014h]
        fldz
        fcompd  word ptr [bp-014h]
        call    cfcc
        jbe     word ptr L296
        fldd    word ptr [bp-014h]
        fchs
        fstpd   word ptr [bp-014h]
L296:
        fldd    word ptr [bp-02ch]
        frsubd  word ptr [bp-01ch]
        fstpd   word ptr [bp-01ch]
        fldz
        fcompd  word ptr [bp-01ch]
        call    cfcc
        jbe     word ptr L297
        fldd    word ptr [bp-01ch]
        fchs
        fstpd   word ptr [bp-01ch]
L297:
        fldd    word ptr [bp-02ch]
        frsubd  word ptr [bp-024h]
        fstpd   word ptr [bp-024h]
        fldz
        fcompd  word ptr [bp-024h]
        call    cfcc
        jbe     word ptr L298
        fldd    word ptr [bp-024h]
        fchs
        fstpd   word ptr [bp-024h]
L298:
        fldd    cs:word ptr L10005
        faddd   word ptr [bp-03ch]
        fstpd   word ptr [bp-084h]
        mov     di, word ptr [bp-06h]
        sal     di, 02h
        add     di, word ptr weight_factor
        fldd    cs:word ptr L10009
```

```
         fdivd    word ptr [bp-084h]
         fstpf    word ptr [di]
         mov      di, word ptr [bp-06h]
         sal      di, 02h
         add      di, word ptr weight_factor
         fldf     word ptr [di]
         mov      di, word ptr [bp-06h]
         sal      di, 02h
         add      di, word ptr weight_factor
         fmulf    word ptr [di]
         fstpf    word ptr [di]
         fldi     word ptr grey_dist
         fcompd   word ptr [bp-014h]
         call     cfcc
         jbe      word ptr L299
         fldi     word ptr grey_dist
         fcompd   word ptr [bp-01ch]
         call     cfcc
         jbe      word ptr L299
         fldi     word ptr grey_dist
         fcompd   word ptr [bp-024h]
         call     cfcc
         jbe      word ptr L299
         mov      di, word ptr [bp-06h]
         sal      di, 02h
         add      di, word ptr weight_factor
         fldi     word ptr grey_factor
         fmulf    word ptr [di]
         fstpf    word ptr [di]
L299:
         mov      word ptr [bp-08h], 00h
L302:
         cmp      word ptr [bp-08h], 04h
         jge      word ptr L300
         mov      di, word ptr [bp-06h]
         sal      di, 02h
         add      di, word ptr weight_factor
         fldf     word ptr [di]
         mov      di, word ptr [bp-08h]
         add      di, word ptr [bp-04h]
         sal      di, 02h
         add      di, word ptr W
         fmulf    word ptr [di]
         fstpf    word ptr [di]
         inc      word ptr [bp-08h]
```

```
          jmp     word ptr L302
L300:
          add     word ptr [bp-04h], 04h
          inc     word ptr [bp-06h]
          jmp     word ptr L282
L280:
          mov     word ptr [bp-08h], 00h
L305:
          cmp     word ptr [bp-08h], 03h
          jge     word ptr L303
          mov     di, word ptr [bp-08h]
          sal     di, 03h
          add     di, bp
          fldi    word ptr rgb_listlength
          frdivd  word ptr [di-05ch]
          fstpd   word ptr [di-05ch]
          inc     word ptr [bp-08h]
          jmp     word ptr L305
L303:
          mov     word ptr [bp-08h], 00h
L308:
          cmp     word ptr [bp-08h], 04h
          jge     word ptr L306
          mov     di, word ptr [bp-08h]
          sal     di, 03h
          add     di, bp
          fldi    word ptr rgb_listlength
          frdivd  word ptr [di-07ch]
          fstpd   word ptr [di-07ch]
          inc     word ptr [bp-08h]
          jmp     word ptr L308
L306:
          mov     di, word ptr [bp-06h]
          sal     di, 02h
          add     di, word ptr weight_factor
          fldd    cs:word ptr L10010
          fstpf   word ptr [di]
          mov     di, word ptr [bp-04h]
          sal     di, 02h
          add     di, word ptr W
          fld1
          fstpf   word ptr [di]
          mov     di, word ptr [bp-04h]
          inc     di
          sal     di, 02h
```

```
            add     di, word ptr W
            fldi    cs:word ptr L10011
            faddd   word ptr [bp-05ch]
            fstpf   word ptr [di]
            mov     di, word ptr [bp-04h]
            inc     di
            inc     di
            sal     di, 02h
            add     di, word ptr W
            fldd    word ptr [bp-054h]
            fstpf   word ptr [di]
            mov     di, word ptr [bp-04h]
            add     di, 03h
            sal     di, 02h
            add     di, word ptr W
            fldd    word ptr [bp-04ch]
            fstpf   word ptr [di]
            mov     word ptr [bp-08h], 00h
L311:
            cmp     word ptr [bp-08h], 04h
            jge     word ptr L309
            mov     di, word ptr [bp-06h]
            sal     di, 02h
            add     di, word ptr weight_factor
            fldf    word ptr [di]
            mov     di, word ptr [bp-08h]
            add     di, word ptr [bp-04h]
            sal     di, 02h
            add     di, word ptr W
            fmulf   word ptr [di]
            fstpf   word ptr [di]
            inc     word ptr [bp-08h]
            jmp     word ptr L311
L309:
            mov     word ptr [bp-08h], 00h
L314:
            cmp     word ptr [bp-08h], 04h
            jge     word ptr L312
            mov     di, word ptr [bp-08h]
            sal     di, 03h
            add     di, bp
            fldd    word ptr [di-07ch]
            mov     si, word ptr [bp-08h]
            sal     si, 02h
            mov     di, word ptr [bp-06h]
```

```
                sal     di, 04h
                add     di, si
                fstpf   word ptr [di+cmyk_list]
                inc     word ptr [bp-08h]
                jmp     word ptr L314
        L312:
                mov     di, word ptr [bp-06h]
                sal     di, 04h
                fldi    cs:word ptr L10011
                faddf   word ptr [di+cmyk_list]
                fstpf   word ptr [di+cmyk_list]
                add     word ptr [bp-04h], 04h
                inc     word ptr [bp-06h]
                mov     di, word ptr [bp-04h]
                sal     di, 02h
                add     di, word ptr W
                fld1
                fstpf   word ptr [di]
                mov     di, word ptr [bp-04h]
                inc     di
                sal     di, 02h
                add     di, word ptr W
                fldd    word ptr [bp-05ch]
                fstpf   word ptr [di]
                mov     di, word ptr [bp-04h]
                inc     di
                inc     di
                sal     di, 02h
                add     di, word ptr W
                fldi    cs:word ptr L10011
                faddd   word ptr [bp-054h]
                fstpf   word ptr [di]
                mov     di, word ptr [bp-04h]
                add     di, 03h
                sal     di, 02h
                add     di, word ptr W
                fldd    word ptr [bp-04ch]
                fstpf   word ptr [di]
                mov     di, word ptr [bp-06h]
                sal     di, 02h
                add     di, word ptr weight_factor
                fldd    cs:word ptr L10010
                fstpf   word ptr [di]
                mov     word ptr [bp-08h], 00h
        L317:
```

```
        cmp     word ptr [bp-08h], 04h
        jge     word ptr L315
        mov     di, word ptr [bp-06h]
        sal     di, 02h
        add     di, word ptr weight_factor
        fldf    word ptr [di]
        mov     di, word ptr [bp-08h]
        add     di, word ptr [bp-04h]
        sal     di, 02h
        add     di, word ptr W
        fmulf   word ptr [di]
        fstpf   word ptr [di]
        inc     word ptr [bp-08h]
        jmp     word ptr L317
L315:
        mov     word ptr [bp-08h], 00h
L320:
        cmp     word ptr [bp-08h], 04h
        jge     word ptr L318
        mov     di, word ptr [bp-08h]
        sal     di, 03h
        add     di, bp
        fldd    word ptr [di-07ch]
        mov     si, word ptr [bp-08h]
        sal     si, 02h
        mov     di, word ptr [bp-06h]
        sal     di, 04h
        add     di, si
        fstpf   word ptr [di+cmyk_list]
        inc     word ptr [bp-08h]
        jmp     word ptr L320
L318:
        mov     di, word ptr [bp-06h]
        sal     di, 04h
        fldi    cs:word ptr L10011
        faddf   word ptr [di+cmyk_list+04h]
        fstpf   word ptr [di+cmyk_list+04h]
        add     word ptr [bp-04h], 04h
        inc     word ptr [bp-06h]
        mov     di, word ptr [bp-04h]
        sal     di, 02h
        add     di, word ptr W
        fld1
        fstpf   word ptr [di]
        mov     di, word ptr [bp-04h]
```

```
        inc     di
        sal     di, 02h
        add     di, word ptr W
        fldd    word ptr [bp-05ch]
        fstpf   word ptr [di]
        mov     di, word ptr [bp-04h]
        inc     di
        inc     di
        sal     di, 02h
        add     di, word ptr W
        fldd    word ptr [bp-054h]
        fstpf   word ptr [di]
        mov     di, word ptr [bp-04h]
        add     di, 03h
        sal     di, 02h
        add     di, word ptr W
        fldi    cs:word ptr L10011
        faddd   word ptr [bp-04ch]
        fstpf   word ptr [di]
        mov     di, word ptr [bp-06h]
        sal     di, 02h
        add     di, word ptr weight_factor
        fldd    cs:word ptr L10010
        fstpf   word ptr [di]
        mov     word ptr [bp-08h], 00h
L323:
        cmp     word ptr [bp-08h], 04h
        jge     word ptr L321
        mov     di, word ptr [bp-06h]
        sal     di, 02h
        add     di, word ptr weight_factor
        fldf    word ptr [di]
        mov     di, word ptr [bp-08h]
        add     di, word ptr [bp-04h]
        sal     di, 02h
        add     di, word ptr W
        fmulf   word ptr [di]
        fstpf   word ptr [di]
        inc     word ptr [bp-08h]
        jmp     word ptr L323
L321:
        mov     word ptr [bp-08h], 00h
L326:
        cmp     word ptr [bp-08h], 04h
        jge     word ptr L324
```

```
        mov     di, word ptr [bp-08h]
        sal     di, 03h
        add     di, bp
        fldd    word ptr [di-07ch]
        mov     si, word ptr [bp-08h]
        sal     si, 02h
        mov     di, word ptr [bp-06h]
        sal     di, 04h
        add     di, si
        fstpf   word ptr [di+cmyk_list]
        inc     word ptr [bp-08h]
        jmp     word ptr L326
L324:
        mov     di, word ptr [bp-06h]
        sal     di, 04h
        fldi    cs:word ptr L10011
        faddf   word ptr [di+cmyk_list+08h]
        fstpf   word ptr [di+cmyk_list+08h]
        add     word ptr rgb_listlength, 03h
        fwait
        leave
        pop     di
        pop     si
        ret
        public getmat
getmat:

        strings

L348:
        db      0ah
        db      020h
        db      07ah
        db      073h
        db      06fh
        db      06ch
        db      076h
        db      065h
        db      020h
        db      065h
        db      072h
        db      072h
        db      06fh
        db      072h
        db      020h
```

```
            db      02dh
            db      020h
            db      063h
            db      06fh
            db      064h
            db      065h
            db      020h
            db      03dh
            db      020h
            db      025h
            db      064h
            db      00h

            code

            push    si
            push    di
            enter   0d0h, 00h
            mov     word ptr [bp-02h], 00h
L330:
            cmp     word ptr [bp-02h], 04h
            jl      word ptr L20109
            jmp     word ptr L328
L20109:
            mov     ax, word ptr [bp-02h]
            mov     word ptr [bp-04h], ax
L333:
            cmp     word ptr [bp-04h], 04h
            jl      word ptr L20110
            jmp     word ptr L331
L20110:
            fldz
            fstpd   word ptr [bp-090h]
            mov     word ptr [bp-06h], 00h
L336:
            mov     ax, word ptr [bp-06h]
            cmp     ax, word ptr rgb_listlength
            jge     word ptr L334
            mov     si, word ptr [bp-06h]
            sal     si, 02h
            add     si, word ptr [bp-04h]
            sal     si, 02h
            add     si, word ptr W
            mov     di, word ptr [bp-06h]
            sal     di, 02h
```

```
        add     di, word ptr [bp-02h]
        sal     di, 02h
        add     di, word ptr W
        fldf    word ptr [di]
        fmulf   word ptr [si]
        faddd   word ptr [bp-090h]
        fstpd   word ptr [bp-090h]
        inc     word ptr [bp-06h]
        jmp     word ptr L336
L334:
        mov     di, word ptr [bp-02h]
        sal     di, 02h
        add     di, word ptr [bp-04h]
        sal     di, 03h
        add     di, bp
        fldd    word ptr [bp-090h]
        fstpd   word ptr [di-088h]
        mov     ax, word ptr [bp-04h]
        cmp     ax, word ptr [bp-02h]
        jle     word ptr L337
        mov     di, word ptr [bp-04h]
        sal     di, 02h
        add     di, word ptr [bp-02h]
        sal     di, 03h
        add     di, bp
        fldd    word ptr [bp-090h]
        fstpd   word ptr [di-088h]
L337:
        inc     word ptr [bp-04h]
        jmp     word ptr L333
L331:
        inc     word ptr [bp-02h]
        jmp     word ptr L330
L328:
        mov     word ptr [bp-02h], 00h
L340:
        cmp     word ptr [bp-02h], 04h
        jl      word ptr L20111
        jmp     word ptr L338
L20111:
        mov     word ptr [bp-04h], 00h
L343:
        cmp     word ptr [bp-04h], 04h
        jge     word ptr L341
        fldz
```

```
            fstpd   word ptr [bp-090h]
            mov     word ptr [bp-06h], 00h
L346:
            mov     ax, word ptr [bp-06h]
            cmp     ax, word ptr rgb_listlength
            jge     word ptr L344
            mov     bx, word ptr [bp-06h]
            sal     bx, 02h
            add     bx, word ptr weight_factor
            mov     di, word ptr [bp-02h]
            sal     di, 02h
            mov     si, word ptr [bp-06h]
            sal     si, 04h
            add     si, di
            mov     di, word ptr [bp-06h]
            sal     di, 02h
            add     di, word ptr [bp-04h]
            sal     di, 02h
            add     di, word ptr W
            fldf    word ptr [di]
            fmulf   word ptr [si+cmyk_list]
            fmulf   word ptr [bx]
            faddd   word ptr [bp-090h]
            fstpd   word ptr [bp-090h]
            inc     word ptr [bp-06h]
            jmp     word ptr L346
L344:
            mov     di, word ptr [bp-04h]
            sal     di, 03h
            add     di, bp
            fldd    word ptr [bp-090h]
            fstpd   word ptr [di-0b0h]
            inc     word ptr [bp-04h]
            jmp     word ptr L343
L341:
            lea     ax, word ptr [bp-0d0h]
            push    ax
            push    04h
            lea     ax, word ptr [bp-0b0h]
            push    ax
            lea     ax, word ptr [bp-088h]
            push    ax
            call    zsolve
            add     sp, 08h
            mov     word ptr [bp-08h], ax
```

```
            cmp     word ptr [bp-08h], 00h
            jge     word ptr L347
            push    ax
            push    offset L348
            call    printf
            add     sp, 04h
            mov     ax, -0ah
            jmp     word ptr L327
L347:
            mov     word ptr [bp-04h], 00h
L351:
            cmp     word ptr [bp-04h], 04h
            jge     word ptr L349
            mov     di, word ptr [bp-04h]
            sal     di, 03h
            add     di, bp
            fldd    word ptr [di-0d0h]
            mov     si, word ptr [bp-04h]
            sal     si, 03h
            mov     di, word ptr [bp-02h]
            sal     di, 05h
            add     di, si
            fstpd   word ptr [di+matrix]
            inc     word ptr [bp-04h]
            jmp     word ptr L351
L349:
            inc     word ptr [bp-02h]
            jmp     word ptr L340
L338:
            sub     ax, ax
L327:
            fwait
            leave
            pop     di
            pop     si
            ret
            public getfew
getfew:
            push    si
            push    di
            enter   03ah, 00h
            mov     word ptr [bp-0ah], 03h
            cmp     word ptr k_flag, 00h
            je      word ptr L353
            mov     word ptr [bp-0ah], 04h
```

```
L353:
        mov     word ptr [bp-04h], 00h
L356:
        cmp     word ptr [bp-04h], 04h
        jge     word ptr L354
        mov     di, word ptr [bp-04h]
        sal     di, 03h
        add     di, bp
        fldz
        fstpd   word ptr [di-032h]
        inc     word ptr [bp-04h]
        jmp     word ptr L356
L354:
        cmp     word ptr k_flag, 00h
        jne     word ptr L357
        fldi    cs:word ptr L10008
        fstpd   word ptr [bp-01ah]
L357:
        mov     word ptr [bp-02h], 00h
L360:
        mov     ax, word ptr [bp-02h]
        cmp     ax, word ptr rgb_listlength
        jl      word ptr L20112
        jmp     word ptr L358
L20112:
        mov     di, word ptr [bp-02h]
        sal     di, 01h
        add     di, word ptr rgb_list
        mov     ax, word ptr [di]
        mov     word ptr [bp-08h], ax
        dec     ax
        imul    word ptr [bp-0ah]
        cwd
        idiv    cs:word ptr L10003
        mov     word ptr [bp-06h], ax
        mov     word ptr [bp-04h], 00h
L363:
        mov     ax, word ptr [bp-04h]
        cmp     ax, word ptr [bp-0ah]
        jge     word ptr L361
        mov     di, word ptr [bp-04h]
        add     di, word ptr [bp-06h]
        sal     di, 03h
        add     di, word ptr A
        fldd    word ptr [di]
```

```
                mov     si, word ptr [bp-04h]
                sal     si, 02h
                mov     di, word ptr [bp-02h]
                sal     di, 04h
                add     di, si
                fstpf   word ptr [di+cmyk_list]
                mov     di, word ptr [bp-04h]
                add     di, word ptr [bp-06h]
                sal     di, 03h
                add     di, word ptr A
                fldd    word ptr [di]
                mov     di, word ptr [bp-04h]
                sal     di, 03h
                add     di, bp
                faddd   word ptr [di-032h]
                fstpd   word ptr [di-032h]
                inc     word ptr [bp-04h]
                jmp     word ptr L363
        L361:
                cmp     word ptr k_flag, 00h
                jne     word ptr L364
                mov     di, word ptr [bp-02h]
                sal     di, 04h
                fldi    cs:word ptr L10008
                fstpf   word ptr [di+cmyk_list+0ch]
        L364:
                inc     word ptr [bp-02h]
                jmp     word ptr L360
        L358:
                fldi    word ptr rgb_listlength
                fstpd   word ptr [bp-03ah]
                fldd    cs:word ptr L10005
                fdivd   word ptr [bp-03ah]
                fstpd   word ptr [bp-012h]
                mov     word ptr [bp-04h], 00h
        L367:
                mov     ax, word ptr [bp-04h]
                cmp     ax, word ptr [bp-0ah]
                jge     word ptr L365
                mov     di, word ptr [bp-04h]
                sal     di, 03h
                add     di, bp
                fldd    word ptr [bp-012h]
                fmuld   word ptr [di-032h]
                fstpd   word ptr [di-032h]
```

```
            inc      word ptr [bp-04h]
            jmp      word ptr L367
L365:
            mov      word ptr [bp-02h], 00h
L370:
            cmp      word ptr [bp-02h], 04h
            jge      word ptr L352
            mov      di, word ptr [bp-02h]
            sal      di, 03h
            add      di, bp
            fldd     word ptr [di-032h]
            mov      di, word ptr [bp-02h]
            sal      di, 05h
            fstpd    word ptr [di+matrix]
            mov      word ptr [bp-04h], 01h
L373:
            cmp      word ptr [bp-04h], 04h
            jge      word ptr L371
            mov      si, word ptr [bp-04h]
            sal      si, 03h
            mov      di, word ptr [bp-02h]
            sal      di, 05h
            add      di, si
            fldz
            fstpd    word ptr [di+matrix]
            inc      word ptr [bp-04h]
            jmp      word ptr L373
L371:
            inc      word ptr [bp-02h]
            jmp      word ptr L370
L352:
            fwait
            leave
            pop      di
            pop      si
            ret
            public getlut
getlut:
            push     si
            push     di
            enter    062h, 00h
            fldd     cs:word ptr L10005
            fstpd    word ptr [bp-03ah]
            mov      si, word ptr [bp+0ah]
            sal      si, 01h
```

```
          mov     di, word ptr [bp+08h]
          mov     ax, word ptr [di]
          sub     ax, word ptr [si+g_range]
          mov     word ptr [bp-02h], ax
L377:
          mov     si, word ptr [bp+0ah]
          sal     si, 01h
          mov     di, word ptr [bp+08h]
          mov     ax, word ptr [di]
          add     ax, word ptr [si+g_range]
          cmp     ax, word ptr [bp-02h]
          jge     word ptr L20114
          jmp     word ptr L374
L20114:
          cmp     word ptr [bp-02h], 00h
          jge     word ptr L20115
          jmp     word ptr L378
L20115:
          cmp     word ptr [bp-02h], 0100h
          jle     word ptr L20116
          jmp     word ptr L378
L20116:
          mov     ax, word ptr [bp-02h]
          sar     ax, 04h
          mov     word ptr [bp-08h], ax
          fldi    word ptr [bp-02h]
          fstpd   word ptr [bp-032h]
          mov     si, word ptr [bp+0ah]
          sal     si, 01h
          mov     ax, word ptr [di+02h]
          sub     ax, word ptr [si+g_range]
          mov     word ptr [bp-04h], ax
L381:
          mov     si, word ptr [bp+0ah]
          sal     si, 01h
          mov     di, word ptr [bp+08h]
          mov     ax, word ptr [di+02h]
          add     ax, word ptr [si+g_range]
          cmp     ax, word ptr [bp-04h]
          jge     word ptr L20117
          jmp     word ptr L378
L20117:
          cmp     word ptr [bp-04h], 00h
          jge     word ptr L20118
          jmp     word ptr L382
```

```
L20118:
        cmp     word ptr [bp-04h], 0100h
        jle     word ptr L20119
        jmp     word ptr L382
L20119:
        mov     ax, word ptr [bp-04h]
        sar     ax, 04h
        mov     word ptr [bp-0ah], ax
        fldi    word ptr [bp-04h]
        fstpd   word ptr [bp-02ah]
        mov     si, word ptr [bp+0ah]
        sal     si, 01h
        mov     ax, word ptr [di+04h]
        sub     ax, word ptr [si+g_range]
        mov     word ptr [bp-06h], ax
L385:
        mov     si, word ptr [bp+0ah]
        sal     si, 01h
        mov     di, word ptr [bp+08h]
        mov     ax, word ptr [di+04h]
        add     ax, word ptr [si+g_range]
        cmp     ax, word ptr [bp-06h]
        jge     word ptr L20120
        jmp     word ptr L382
L20120:
        cmp     word ptr [bp-06h], 00h
        jge     word ptr L20121
        jmp     word ptr L386
L20121:
        cmp     word ptr [bp-06h], 0100h
        jle     word ptr L20122
        jmp     word ptr L386
L20122:
        mov     ax, word ptr [bp-06h]
        sar     ax, 04h
        mov     word ptr [bp-0ch], ax
        fldi    word ptr [bp-06h]
        fstpd   word ptr [bp-022h]
        imul    ax, word ptr [bp-0ah], 011h
        mov     di, ax
        imul    ax, word ptr [bp-0ch], 0121h
        add     ax, di
        add     ax, word ptr [bp-08h]
        mov     word ptr [bp-012h], ax
        mov     di, word ptr [bp+08h]
```

```
        mov     ax, word ptr [di]
        sub     ax, word ptr [bp-02h]
        mov     word ptr [bp-018h], ax
        mov     ax, word ptr [di+02h]
        sub     ax, word ptr [bp-04h]
        mov     word ptr [bp-016h], ax
        mov     ax, word ptr [di+04h]
        sub     ax, word ptr [bp-06h]
        mov     word ptr [bp-014h], ax
        mov     word ptr [bp-010h], 00h
L389:
        cmp     word ptr [bp-010h], 03h
        jge     word ptr L387
        mov     di, word ptr [bp-010h]
        sal     di, 01h
        add     di, bp
        cmp     word ptr [di-018h], 00h
        jge     word ptr L390
        mov     di, word ptr [bp-010h]
        sal     di, 01h
        add     di, bp
        mov     ax, word ptr [di-018h]
        neg     ax
        mov     di, word ptr [bp-010h]
        sal     di, 01h
        add     di, bp
        mov     word ptr [di-018h], ax
L390:
        mov     di, word ptr [bp-010h]
        sal     di, 01h
        add     di, bp
        mov     ax, word ptr [di-018h]
        sar     ax, 04h
        mov     di, word ptr [bp-010h]
        sal     di, 01h
        add     di, bp
        mov     word ptr [di-018h], ax
        inc     word ptr [bp-010h]
        jmp     word ptr L389
L387:
        mov     ax, word ptr [bp-018h]
        cmp     ax, word ptr [bp-016h]
        jg      word ptr L20113
        mov     ax, word ptr [bp-016h]
L20113:
```

```
        mov     word ptr [bp-01ah], ax
        mov     ax, word ptr [bp-014h]
        cmp     ax, word ptr [bp-01ah]
        jle     word ptr L393
        mov     word ptr [bp-01ah], ax
L393:
        mov     si, word ptr [bp+0ah]
        sal     si, 03h
        mov     di, word ptr [bp-01ah]
        sal     di, 05h
        add     di, si
        fldd    word ptr [di+f2]
        fstpd   word ptr [bp-062h]
        fldd    word ptr [bp-062h]
        fcompd  word ptr epsilon
        call    cfcc
        jae     word ptr L20123
        jmp     word ptr L386
L20123:
        mov     di, word ptr [bp-012h]
        sal     di, 03h
        add     di, word ptr weight_vec
        fldd    word ptr [bp-062h]
        faddd   word ptr [di]
        fstpd   word ptr [di]
        mov     word ptr [bp-0eh], 00h
L398:
        cmp     word ptr [bp-0eh], 04h
        jl      word ptr L20124
        jmp     word ptr L396
L20124:
        mov     di, word ptr [bp-0eh]
        sal     di, 03h
        add     di, bp
        fldz
        fstpd   word ptr [di-05ah]
        mov     word ptr [bp-010h], 00h
L401:
        cmp     word ptr [bp-010h], 04h
        jge     word ptr L399
        mov     bx, word ptr [bp-010h]
        sal     bx, 03h
        add     bx, bp
        mov     si, word ptr [bp-010h]
        sal     si, 03h
```

```
            mov     di, word ptr [bp-0eh]
            sal     di, 05h
            add     di, si
            fldd    word ptr [di+matrix]
            fmuld   word ptr [bx-03ah]
            mov     di, word ptr [bp-0eh]
            sal     di, 03h
            add     di, bp
            faddd   word ptr [di-05ah]
            fstpd   word ptr [di-05ah]
            inc     word ptr [bp-010h]
            jmp     word ptr L401
L399:
            mov     di, word ptr [bp-0eh]
            sal     di, 03h
            add     di, bp
            fldi    word ptr lowlimit
            fcompd  word ptr [di-05ah]
            call    cfcc
            jbe     word ptr L402
            mov     di, word ptr [bp-0eh]
            sal     di, 03h
            add     di, bp
            fldi    word ptr lowlimit
            fstpd   word ptr [di-05ah]
L402:
            mov     di, word ptr [bp-0eh]
            sal     di, 03h
            add     di, bp
            fldi    word ptr highlimit
            fcompd  word ptr [di-05ah]
            call    cfcc
            jae     word ptr L403
            mov     di, word ptr [bp-0eh]
            sal     di, 03h
            add     di, bp
            fldi    word ptr highlimit
            fstpd   word ptr [di-05ah]
L403:
            inc     word ptr [bp-0eh]
            jmp     word ptr L398
L396:
            mov     di, word ptr [bp-012h]
            sal     di, 03h
            add     di, word ptr clut
```

```
          fldd    word ptr [bp-062h]
          fmuld   word ptr [bp-05ah]
          faddd   word ptr [di]
          fstpd   word ptr [di]
          mov     di, word ptr [bp-012h]
          sal     di, 03h
          add     di, word ptr mlut
          fldd    word ptr [bp-062h]
          fmuld   word ptr [bp-052h]
          faddd   word ptr [di]
          fstpd   word ptr [di]
          mov     di, word ptr [bp-012h]
          sal     di, 03h
          add     di, word ptr ylut
          fldd    word ptr [bp-062h]
          fmuld   word ptr [bp-04ah]
          faddd   word ptr [di]
          fstpd   word ptr [di]
          mov     di, word ptr [bp-012h]
          sal     di, 03h
          add     di, word ptr klut
          fldd    word ptr [bp-062h]
          fmuld   word ptr [bp-042h]
          faddd   word ptr [di]
          fstpd   word ptr [di]
L386:
          add     word ptr [bp-06h], 010h
          jmp     word ptr L385
L382:
          add     word ptr [bp-04h], 010h
          jmp     word ptr L381
L378:
          add     word ptr [bp-02h], 010h
          jmp     word ptr L377
L374:
          fwait
          leave
          pop     di
          pop     si
          ret
          public growtree
growtree:

          strings
```

```
L427:
        db      020h
        db      0ah
        db      020h
        db      074h
        db      072h
        db      065h
        db      065h
        db      020h
        db      06fh
        db      076h
        db      065h
        db      072h
        db      066h
        db      06ch
        db      06fh
        db      077h
        db      020h
        db      00h

        code

        push    si
        push    di
        enter   016h, 00h
        mov     di, word ptr [bp+08h]
        mov     ax, word ptr [di+02h]
        mov     word ptr [bp-0eh], ax
        cmp     word ptr [bp-0eh], 07h
        jle     word ptr L405
L20129:
        sub     ax, ax
        jmp     word ptr L404
L405:
        mov     word ptr [bp-014h], 00h
L408:
        cmp     word ptr [bp-014h], 03h
        jl      word ptr L20130
        jmp     word ptr L406
L20130:
        mov     di, word ptr [bp-014h]
        add     di, word ptr [bp+0ah]
        sal     di, 03h
        add     di, word ptr list
        fldd    word ptr [di]
```

```
        faddd   cs:word ptr L10006
        call    dp87
        call    idcvt
        add     sp, 08h
        mov     di, word ptr [bp-014h]
        sal     di, 01h
        add     di, bp
        mov     word ptr [di-06h], ax
        mov     di, word ptr [bp-014h]
        sal     di, 01h
        add     di, bp
        mov     ax, word ptr [di-06h]
        cmp     ax, word ptr maxthresh
        jg      word ptr L20129
        mov     di, word ptr [bp-014h]
        sal     di, 01h
        add     di, bp
        mov     ax, word ptr [di-06h]
        cmp     ax, word ptr minthresh
        jl      word ptr L20129
        mov     di, word ptr [bp-014h]
        sal     di, 01h
        add     di, bp
        mov     ax, word ptr [di-06h]
        cmp     ax, 0ffh
        jle     word ptr L410
        mov     di, word ptr [bp-014h]
        sal     di, 01h
        add     di, bp
        mov     word ptr [di-06h], 0ffh
L410:
        mov     di, word ptr [bp-014h]
        sal     di, 01h
        add     di, bp
        cmp     word ptr [di-06h], 00h
        jge     word ptr L411
        mov     di, word ptr [bp-014h]
        sal     di, 01h
        add     di, bp
        mov     word ptr [di-06h], 00h
L411:
        inc     word ptr [bp-014h]
        jmp     word ptr L408
L406:
        mov     ax, 07h
```

```
          sub     ax, word ptr [bp-0eh]
          mov     word ptr [bp-010h], ax
L414:
          cmp     word ptr [bp-010h], 00h
          jl      word ptr L412
          mov     word ptr [bp-012h], 00h
          mov     word ptr [bp-014h], 00h
L417:
          cmp     word ptr [bp-014h], 03h
          jge     word ptr L415
          mov     di, word ptr [bp-014h]
          sal     di, 01h
          add     di, bp
          mov     ax, word ptr [di-06h]
          mov     cx, word ptr [bp-010h]
          sar     ax, cl
          and     ax, 01h
          mov     cx, word ptr [bp-014h]
          sal     ax, cl
          add     word ptr [bp-012h], ax
          inc     word ptr [bp-014h]
          jmp     word ptr L417
L415:
          mov     di, word ptr [bp-012h]
          sal     di, 01h
          add     di, word ptr [bp+08h]
          cmp     word ptr [di+04h], 00h
          jle     word ptr L412
          mov     di, word ptr [bp-012h]
          sal     di, 01h
          add     di, word ptr [bp+08h]
          imul    ax, word ptr [di+04h], 014h
          add     ax, word ptr root
          mov     word ptr [bp+08h], ax
          dec     word ptr [bp-010h]
          jmp     word ptr L414
L412:
          mov     di, word ptr [bp+08h]
          mov     ax, word ptr [di+02h]
          mov     word ptr [bp-0eh], ax
          mov     di, word ptr [bp-012h]
          sal     di, 01h
          add     di, word ptr [bp+08h]
          cmp     word ptr [di+04h], 00h
          jne     word ptr L420
```

```
            mov     di, word ptr [bp-012h]
            sal     di, 01h
            add     di, word ptr [bp+08h]
            mov     ax, word ptr [bp+0ah]
            neg     ax
            mov     word ptr [di+04h], ax
            jmp     word ptr L20129
L420:
            mov     di, word ptr [bp-012h]
            sal     di, 01h
            add     di, word ptr [bp+08h]
            mov     ax, word ptr [di+04h]
            neg     ax
            mov     word ptr [bp-016h], ax
            mov     word ptr [bp-014h], 00h
L424:
            cmp     word ptr [bp-014h], 03h
            jge     word ptr L422
            mov     di, word ptr [bp-014h]
            add     di, word ptr [bp-016h]
            sal     di, 03h
            add     di, word ptr list
            fldd    word ptr [di]
            faddd   cs:word ptr L10006
            call    dp87
            call    idcvt
            add     sp, 08h
            mov     di, word ptr [bp-014h]
            sal     di, 01h
            add     di, bp
            mov     word ptr [di-0ch], ax
            inc     word ptr [bp-014h]
            jmp     word ptr L424
L422:
            mov     ax, word ptr [bp-06h]
            cmp     ax, word ptr [bp-0ch]
            jne     word ptr L425
            mov     ax, word ptr [bp-04h]
            cmp     ax, word ptr [bp-0ah]
            jne     word ptr L425
            mov     ax, word ptr [bp-02h]
            cmp     ax, word ptr [bp-08h]
            jne     word ptr L425
            jmp     word ptr L20129
L425:
```

```
        inc     word ptr treelength
        mov     ax, word ptr treelength
        cmp     ax, word ptr msize
        jle     word ptr L426
        push    offset L427
        call    printf
        add     sp, 02h
        jmp     word ptr L20126
L426:
        mov     di, word ptr [bp-012h]
        sal     di, 01h
        add     di, word ptr [bp+08h]
        mov     ax, word ptr treelength
        mov     word ptr [di+04h], ax
        imul    ax, word ptr treelength, 014h
        add     ax, word ptr root
        mov     word ptr [bp+08h], ax
        mov     ax, word ptr [bp-0eh]
        inc     ax
        mov     di, word ptr [bp+08h]
        mov     word ptr [di+02h], ax
        push    word ptr [bp-016h]
        push    di
        call    growtree
        add     sp, 04h
        or      ax, ax
        jl      word ptr L20126
        push    word ptr [bp+0ah]
        push    word ptr [bp+08h]
        call    growtree
        add     sp, 04h
        or      ax, ax
        jl      word ptr L20126
        jmp     word ptr L20129
L20126:
        mov     ax, -01h
L404:
        fwait
        leave
        pop     di
        pop     si
        ret
        public number_of_sons
number_of_sons:
        push    si
```

```
            push    di
            enter   06h, 00h
            mov     word ptr [bp-06h], 00h
            mov     word ptr [bp-02h], 00h
L433:
            cmp     word ptr [bp-02h], 08h
            jge     word ptr L431
            mov     di, word ptr [bp-02h]
            sal     di, 01h
            add     di, word ptr [bp+08h]
            mov     ax, word ptr [di+04h]
            mov     word ptr [bp-04h], ax
            cmp     word ptr [bp-04h], 00h
            jge     word ptr L434
            inc     word ptr [bp-06h]
            jmp     word ptr L435
L434:
            cmp     word ptr [bp-04h], 00h
            jle     word ptr L435
            imul    ax, word ptr [bp-04h], 014h
            add     ax, word ptr root
            push    ax
            call    number_of_sons
            add     sp, 02h
            add     word ptr [bp-06h], ax
L435:
            inc     word ptr [bp-02h]
            jmp     word ptr L433
L431:
            mov     ax, word ptr [bp-06h]
            mov     di, word ptr [bp+08h]
            mov     word ptr [di], ax
            mov     ax, word ptr [bp-06h]
            leave
            pop     di
            pop     si
            ret
            public findleaf
findleaf:
            push    si
            push    di
            enter   0ah, 00h
            mov     word ptr [bp-0ah], 00h
            mov     ax, word ptr root
L20131:
```

```
        mov     word ptr [bp-02h], ax
        mov     di, word ptr [bp-02h]
        mov     ax, word ptr [di+02h]
        cmp     ax, word ptr [bp+0ah]
        jge     word ptr L438
        mov     ax, 07h
        sub     ax, word ptr [di+02h]
        mov     word ptr [bp-04h], ax
        mov     word ptr [bp-06h], 00h
        mov     word ptr [bp-08h], 00h
L442:
        cmp     word ptr [bp-08h], 03h
        jge     word ptr L440
        mov     di, word ptr [bp-08h]
        sal     di, 01h
        add     di, word ptr [bp+08h]
        mov     ax, word ptr [di]
        mov     cx, word ptr [bp-04h]
        sar     ax, cl
        and     ax, 01h
        mov     cx, word ptr [bp-08h]
        sal     ax, cl
        add     word ptr [bp-06h], ax
        inc     word ptr [bp-08h]
        jmp     word ptr L442
L440:
        mov     di, word ptr [bp-06h]
        sal     di, 01h
        add     di, word ptr [bp-02h]
        mov     ax, word ptr [di+04h]
        mov     word ptr [bp-0ah], ax
        cmp     word ptr [bp-0ah], 00h
        jle     word ptr L438
        imul    ax, word ptr [bp-0ah], 014h
        add     ax, word ptr root
        jmp     word ptr L20131
L438:
        mov     ax, word ptr [bp-0ah]
        leave
        pop     di
        pop     si
        ret
        public checkval
checkval:
        push    si
```

```
            push    di
            enter   06h, 00h
            mov     di, 08h
            sub     di, word ptr [bp+0ah]
            mov     ax, -01h
            mov     cx, di
            sal     ax, cl
            xor     ax, -0100h
            mov     word ptr [bp-02h], ax
            mov     word ptr [bp-04h], 01h
            mov     word ptr [bp-06h], 00h
L448:
            cmp     word ptr [bp-06h], 03h
            jge     word ptr L446
            mov     di, word ptr [bp-06h]
            add     di, word ptr [bp+0ch]
            sal     di, 03h
            add     di, word ptr list
            fldd    word ptr [di]
            faddd   cs:word ptr L10006
            call    dp87
            call    idcvt
            add     sp, 08h
            mov     si, ax
            and     si, word ptr [bp-02h]
            mov     di, word ptr [bp-06h]
            sal     di, 01h
            add     di, word ptr [bp+08h]
            mov     ax, word ptr [di]
            and     ax, word ptr [bp-02h]
            cmp     ax, si
            je      word ptr L449
            mov     word ptr [bp-04h], 00h
            jmp     word ptr L446
L449:
            inc     word ptr [bp-06h]
            jmp     word ptr L448
L446:
            mov     ax, word ptr [bp-04h]
            fwait
            leave
            pop     di
            pop     si
            ret
            public readtree
```

```
readtree:
          push    si
          push    di
          enter   04h, 00h
          mov     word ptr [bp-02h], 00h
L453:
          cmp     word ptr [bp-02h], 08h
          jge     word ptr L450
          mov     di, word ptr [bp-02h]
          sal     di, 01h
          add     di, word ptr [bp+08h]
          mov     ax, word ptr [di+04h]
          mov     word ptr [bp-04h], ax
          cmp     word ptr [bp-04h], 00h
          jge     word ptr L454
          mov     di, word ptr rgb_listlength
          inc     word ptr rgb_listlength
          sal     di, 01h
          add     di, word ptr rgb_list
          neg     ax
          mov     word ptr [di], ax
          jmp     word ptr L455
L454:
          cmp     word ptr [bp-04h], 00h
          jle     word ptr L455
          imul    ax, word ptr [bp-04h], 014h
          add     ax, word ptr root
          push    ax
          call    readtree
          add     sp, 02h
L455:
          inc     word ptr [bp-02h]
          jmp     word ptr L453
L450:
          leave
          pop     di
          pop     si
          ret
```

```
        name CRTLUTMAIN
        impure
mrk_fname:
        strings
L1:
        db      06dh
        db      072h
        db      06bh
        db      02eh
        db      073h
        db      063h
        db      072h
        db      00h
        impure
        dw      L1
        bss
        public LIGHT0
LIGHT0:
        db      02h dup(0)
        public LIGHT1
LIGHT1:
        db      02h dup(0)
        public LIGHT2
LIGHT2:
        db      02h dup(0)
        public LIGHT3
LIGHT3:
        db      02h dup(0)
        public DARK0
DARK0:
        db      02h dup(0)
        public DARK1
DARK1:
        db      02h dup(0)
        public DARK2
DARK2:
        db      02h dup(0)
```

```
        public DARK3
DARK3:
        db      02h dup(0)
        public D_12
D_12:
        db      02h dup(0)
        public W_12
W_12:
        db      02h dup(0)
        public GAIN_LEVEL_1
GAIN_LEVEL_1:
        db      02h dup(0)
        public LUT_TYPE
LUT_TYPE:
        db      02h dup(0)
        public buf12to12
buf12to12:
        db      06002h dup(0)
        public grad8bit1
grad8bit1:
        db      0800h dup(0)
scan_tok:
        db      02h dup(0)
scan_token:
        db      02h dup(0)
root_token:
        db      02h dup(0)
scid_token:
        db      02h dup(0)
scanner1_sem:
        db      02h dup(0)
scanner2_sem:
        db      02h dup(0)
        public sc_curnt_sem
sc_curnt_sem:
        db      02h dup(0)
        public sc_other_sem
sc_other_sem:
        db      02h dup(0)
scanner1_rdy:
        db      02h dup(0)
scanner2_rdy:
        db      02h dup(0)
scl_rdy_ptr:
        db      02h dup(0)
```

```
sc2_rdy_ptr:
        db      02h dup(0)
sc_rdy_ptr:
        db      02h dup(0)
scid_pointer:
        db      02h dup(0)
appl_pointer:
        db      02h dup(0)
resp_pointer:
        db      02h dup(0)
obj_token:
        db      02h dup(0)
response_ptr:
        db      02h dup(0)
histogram_fname:
        db      064h dup(0)

        impure

sc1tok:
        db      0ah
        db      053h
        db      043h
        db      041h
        db      04eh
        db      031h
        db      054h
        db      04fh
        db      04bh
        db      00h
sc2tok:
        db      0ah
        db      053h
        db      043h
        db      041h
        db      04eh
        db      032h
        db      054h
        db      04fh
        db      04bh
        db      00h
scidtok:
        db      0ah
        db      053h
        db      043h
```

```
        db      041h
        db      04eh
        db      049h
        db      044h
        db      054h
        db      04fh
        db      04bh
        db      00h
smbx:
        db      0ah
        db      053h
        db      043h
        db      041h
        db      04eh
        db      04dh
        db      041h
        db      049h
        db      04ch
        db      042h
        db      058h
        db      00h
srbx:
        db      0ah
        db      053h
        db      043h
        db      041h
        db      04eh
        db      052h
        db      053h
        db      050h
        db      04dh
        db      042h
        db      058h
        db      00h
lmbx:
        db      0ah
        db      04ch
        db      045h
        db      041h
        db      044h
        db      04dh
        db      041h
        db      049h
        db      04ch
        db      042h
```

```
              db        058h
              db        00h
lrbx:
              db        0ah
              db        04ch
              db        045h
              db        041h
              db        044h
              db        052h
              db        053h
              db        050h
              db        04dh
              db        042h
              db        058h
              db        00h
sprm:
              db        0ah
              db        053h
              db        043h
              db        041h
              db        04eh
              db        05fh
              db        050h
              db        041h
              db        052h
              db        041h
              db        04dh
              db        00h
aplt:
              db        0ah
              db        041h
              db        050h
              db        050h
              db        04ch
              db        05fh
              db        054h
              db        04fh
              db        04bh
              db        045h
              db        04eh
              db        00h
rspt:
              db        0ah
              db        052h
              db        045h
```

```
        db      053h
        db      050h
        db      05fh
        db      054h
        db      04fh
        db      04bh
        db      045h
        db      04eh
        db      00h
sc1sem:
        db      0ah
        db      053h
        db      043h
        db      041h
        db      04eh
        db      045h
        db      052h
        db      031h
        db      053h
        db      045h
        db      04dh
        db      00h
sc2sem:
        db      0ah
        db      053h
        db      043h
        db      041h
        db      04eh
        db      045h
        db      052h
        db      032h
        db      053h
        db      045h
        db      04dh
        db      00h
sc1rdy:
        db      0ah
        db      053h
        db      043h
        db      041h
        db      04eh
        db      045h
        db      052h
        db      031h
        db      052h
```

```
        db      044h
        db      059h
        db      00h
sc2rdy:
        db      0ah
        db      053h
        db      043h
        db      041h
        db      04eh
        db      045h
        db      052h
        db      032h
        db      052h
        db      044h
        db      059h
        db      00h
startup_name:
        db      0ah
        db      053h
        db      054h
        db      052h
        db      054h
        db      055h
        db      050h
        db      054h
        db      04bh
        db      031h
        db      0ah
        db      053h
        db      054h
        db      052h
        db      054h
        db      055h
        db      050h
        db      054h
        db      04bh
        db      032h
interactive_name:
        db      0ah
        db      049h
        db      04eh
        db      054h
        db      052h
        db      043h
        db      054h
```

```
                db      054h
                db      04bh
                db      031h
                db      0ah
                db      049h
                db      04eh
                db      054h
                db      052h
                db      043h
                db      054h
                db      054h
                db      04bh
                db      032h
scan_name:
                db      09h
                db      053h
                db      043h
                db      04eh
                db      054h
                db      04fh
                db      04bh
                db      045h
                db      04eh
                db      031h
                db      09h
                db      053h
                db      043h
                db      04eh
                db      054h
                db      04fh
                db      04bh
                db      045h
                db      04eh
                db      032h
exit_twins_name:
                db      09h
                db      045h
                db      058h
                db      049h
                db      054h
                db      054h
                db      057h
                db      049h
                db      04eh
                db      053h
```

```
        bss

scanner_id:
        db      02h dup(0)
        public s_p
s_p:
        db      0126h dup(0)
        public ps_p
ps_p:
        db      0a0h dup(0)
        public files
files:
        db      010h dup(0)
        public ct_struct
ct_struct:
        db      08c4h dup(0)
        public smart_crt
smart_crt:
        db      096ch dup(0)
        public ct_path
ct_path:
        db      08c2h dup(0)
        public glpar
glpar:
        db      024h dup(0)
        public new_setup_pos
new_setup_pos:
        db      070h dup(0)
        public def_setup_pos
def_setup_pos:
        db      070h dup(0)
        public new_grad
new_grad:
        db      012f7h dup(0)
        public def_grad
def_grad:
        db      012f7h dup(0)
        public lutid
lutid:
        db      02h dup(0)
        public def_lut
def_lut:
        db      02h dup(0)
        public def_rgb12to12
```

```
def_rgb12to12:
        db      02h dup(0)
        public def_LUT_TYPE
def_LUT_TYPE:
        db      02h dup(0)

        impure

        public LUT_TYPE_id
LUT_TYPE_id:
        dw      014h

        bss

        public c_p
c_p:
        db      0a8h dup(0)

        impure

        public exp_num
exp_num:
        db      0cch
        db      0cch
        db      0cch
        db      03eh

        bss

        public gcor_lut
gcor_lut:
        db      02h dup(0)
tok13d:
        db      02h dup(0)
tok23d:
        db      02h dup(0)
def_c_p:
        db      0a8h dup(0)
c_p_id:
        db      0a8h dup(0)
def_exp_num:
        db      04h dup(0)
exp_num_id:
        db      04h dup(0)
        public bef_grad8
```

b

```
ef_grad8:
        db      0800h dup(0)

        code

        public main
main:

        strings

L3:
        db      043h
        db      052h
        db      054h
        db      05fh
        db      04ch
        db      055h
        db      054h
        db      00h

        code

        push    si
        push    di
        enter   02h, 00h
        mov     word ptr [bp-02h], 00h
        push    08c4h
        push    offset ct_struct
        call    unstackj
        add     sp, 04h
        push    08c2h
        push    offset ct_struct+02h
        push    offset ct_path
        call    blkmv
        add     sp, 06h
        mov     ax, word ptr ct_struct
        and     ax, -04000h
        sar     ax, 0eh
        mov     word ptr scanner_id, ax
        mov     word ptr scanner_id, 01h
        mov     ax, 0ffh
        mov     word ptr LIGHT3, ax
        mov     word ptr LIGHT2, ax
        mov     word ptr LIGHT1, ax
        mov     word ptr LIGHT0, ax
```

```
        sub     ax, ax
        mov     word ptr DARK3, ax
        mov     word ptr DARK2, ax
        mov     word ptr DARK1, ax
        mov     word ptr DARK0, ax
        mov     word ptr rgb12to12, offset buf12to12
        call    unit_init
        call    init_disp
        call    InitDmtr
        call    load_smart
        call    end_disp
        mov     word ptr gcor_lut, 00h
        push    offset L3
        call    uimsif
        add     sp, 02h
        leave
        pop     di
        pop     si
        ret
        public load_h_i
load_h_i:

        strings

L8:
        db      020h
        db      04eh
        db      04fh
        db      020h
        db      06dh
        db      065h
        db      06dh
        db      06fh
        db      072h
        db      079h
        db      020h
        db      066h
        db      06fh
        db      072h
        db      020h
        db      031h
        db      032h
        db      074h
        db      06fh
        db      031h
```

```
        db      032h
        db      020h
        db      062h
        db      075h
        db      066h
        db      066h
        db      065h
        db      072h
        db      020h
        db      0ah
        db      00h

        code

        strings

L9:
        db      068h
        db      05fh
        db      069h
        db      00h

        code

        strings

L11:
        db      0ah
        db      020h
        db      045h
        db      04eh
        db      044h
        db      020h
        db      04fh
        db      046h
        db      020h
        db      053h
        db      043h
        db      041h
        db      04eh
        db      020h
        db      02dh
        db      020h
```

```
        db      052h
        db      047h
        db      042h
        db      020h
        db      0ah
        db      00h

        code

        strings

L12:
        db      043h
        db      052h
        db      054h
        db      05fh
        db      04ch
        db      055h
        db      054h
        db      00h

        code

        push    si
        push    di
        enter   02h, 00h
        call    unload_smart
        push    si
        push    di
        push    es
        push    00h
        lea     ax, word ptr [bp-02h]
        push    ax
        call    rqcreatesegment
        pop     es
        pop     di
        pop     si
        mov     word ptr tok13d, ax
        cmp     word ptr [bp-02h], 00h
        je      word ptr L5
        push    00h
        push    00h
        push    00h
        push    word ptr [bp-02h]
```

```
            call    handle_exc
            add     sp, 08h
L5:
            push    00h
            push    word ptr tok13d
            call    buildptr
            add     sp, 04h
            mov     word ptr def_lut, ax
            push    si
            push    di
            push    es
            push    00h
            lea     ax, word ptr [bp-02h]
            push    ax
            call    rqcreatesegment
            pop     es
            pop     di
            pop     si
            mov     word ptr tok23d, ax
            cmp     word ptr [bp-02h], 00h
            je      word ptr L6
            push    00h
            push    00h
            push    00h
            push    word ptr [bp-02h]
            call    handle_exc
            add     sp, 08h
L6:
            push    00h
            push    word ptr tok23d
            call    buildptr
            add     sp, 04h
            mov     word ptr lutid, ax
            push    03001h
            push    02h
            call    calloc
            add     sp, 04h
            mov     word ptr def_rgb12to12, ax
            or      ax, ax
            jne     word ptr L7
            push    offset L8
            call    printf
            jmp     word ptr L20001
L7:
            push    word ptr def_rgb12to12
```

```
push    offset def_LUT_TYPE
push    offset def_exp_num
push    offset def_c_p
push    word ptr def_lut
call    save_lut
add     sp, 0ah
push    offset def_grad
call    save_grad
add     sp, 02h
call    init_grad
push    012f7h
push    offset def_grad
push    offset new_grad
call    blkmv
add     sp, 06h
push    0136h
push    offset int_c_p
push    offset new_grad
call    blkmv
add     sp, 06h
push    offset new_grad
call    load_def_grad
add     sp, 02h
push    offset exp_num_id
push    offset c_p_id
push    word ptr lutid
call    read_lutid
add     sp, 06h
call    init_12to12
push    word ptr rgb12to12
push    offset LUT_TYPE_id
push    offset exp_num_id
push    offset c_p_id
push    word ptr lutid
call    load_def_lut
add     sp, 0ah
push    si
push    di
push    es
push    word ptr tok23d
lea     ax, word ptr [bp-02h]
push    ax
call    rqdeletesegment
pop     es
pop     di
```

```
pop     si
push    00h
push    00h
push    00h
push    0271ah
call    handle_exc
add     sp, 08h
call    abort_create_lut
movb    byte ptr ct_path+057h, 00h
push    08c2h
push    offset ct_path
push    offset ct_struct+02h
call    blkmv
add     sp, 06h
push    08c4h
push    offset ct_struct
push    offset smart_crt
call    blkmv
add     sp, 06h
push    096ch
push    offset smart_crt
push    04h
push    0ah
push    01h
push    offset L9
call    loadjob
add     sp, 0ch
push    08c4h
push    offset smart_crt
push    offset ct_struct
call    blkmv
add     sp, 06h
push    08c2h
push    offset ct_struct+02h
push    offset ct_path
call    blkmv
add     sp, 06h
mov     word ptr ct_struct, 06h
push    word ptr def_rgb12to12
push    offset def_LUT_TYPE
push    offset def_exp_num
push    offset def_c_p
push    word ptr def_lut
call    load_def_lut
add     sp, 0ah
```

```
        push    offset def_grad
        call    load_def_grad
        add     sp, 02h
        push    si
        push    di
        push    es
        push    word ptr tok13d
        lea     ax, word ptr [bp-02h]
        push    ax
        call    rqdeletesegment
        pop     es
        pop     di
        pop     si
        push    word ptr def_rgb12to12
        call    free
        add     sp, 02h
        cmpb    byte ptr ct_path+057h, 00h
        je      word ptr L10
        push    offset L11
        call    printf
        add     sp, 02h
        push    00h
        push    00h
        push    00h
        push    0271bh
        call    handle_exc
        add     sp, 08h
        push    0a8h
        push    offset smart_crt+08c4h
        push    offset c_p
        call    blkmv
        add     sp, 06h
        mov     ax, word ptr c_p+06h
        mov     word ptr WHITE12, ax
        mov     ax, word ptr c_p
        mov     word ptr DARK12, ax
L10:
        call    unit_init
        call    init_disp
        call    InitDmtr
        call    load_smart
        call    end_disp
        lea     ax, word ptr [bp-02h]
        push    ax
        push    013h
```

```
        push    03h
        push    06fh
        call    gksif
        add     sp, 08h
        push    offset L12
        call    uimsif
L20001:
        add     sp, 02h
        leave
        pop     di
        pop     si
        ret
        public calib_ter
calib_ter:
        push    si
        push    di
        enter   02h, 00h
        lea     ax, word ptr [bp-02h]
        push    ax
        push    013h
        push    03h
        push    06fh
        call    gksif
        add     sp, 08h
        lea     ax, word ptr [bp-02h]
        push    ax
        push    03h
        push    03h
        push    06fh
        call    gksif
        add     sp, 08h
        leave
        pop     di
        pop     si
        ret
        public abort_create_lut
abort_create_lut:

        strings

L19:
        db      045h
        db      072h
        db      072h
        db      06fh
```

```
        db      072h
        db      020h
        db      069h
        db      06eh
        db      020h
        db      067h
        db      064h
        db      065h
        db      061h
        db      063h
        db      074h
        db      069h
        db      076h
        db      020h
        db      021h
        db      021h
        db      020h
        db      0ah
        db      00h

        code

        strings

L23:
        db      045h
        db      072h
        db      072h
        db      06fh
        db      072h
        db      020h
        db      069h
        db      06eh
        db      020h
        db      067h
        db      063h
        db      06ch
        db      06fh
        db      073h
        db      067h
        db      06bh
        db      073h
        db      021h
        db      021h
```

```
        db      020h
        db      0ah
        db      00h

        code

        push    si
        push    di
        enter   02h, 00h
        mov     word ptr [bp-02h], 00h
        cmp     word ptr files, 00h
        je      word ptr L15
        push    offset files
        call    CloseFile
        add     sp, 02h
L15:
        call    CloseDmtr
        call    end_bufs
        call    closeuims
        lea     ax, word ptr [bp-02h]
        push    ax
        push    02h
        push    068h
        call    gksif
        add     sp, 06h
        cmp     word ptr [bp-02h], 00h
        je      word ptr L16
        cmpb    byte ptr glpar+023h, 00h
        je      word ptr L16
        push    offset L19
        call    printf
        add     sp, 02h
L16:
        lea     ax, word ptr [bp-02h]
        push    ax
        push    02h
        push    06ah
        call    gksif
        add     sp, 06h
        cmp     word ptr [bp-02h], 00h
        je      word ptr L14
        cmpb    byte ptr glpar+023h, 00h
        je      word ptr L14
        push    offset L23
        call    printf
```

```
        add     sp, 02h
L14:
        leave
        pop     di
        pop     si
        ret
        public exit_job
exit_job:
        push    si
        push    di
        push    bp
        mov     bp, sp
        call    unload_smart
        movb    byte ptr ct_path+057h, 00h
        movb    byte ptr ct_path+0828h, 00h
        call    abort_create_lut
        push    08c2h
        push    offset ct_path
        push    offset ct_struct+02h
        call    blkmv
        add     sp, 06h
        mov     word ptr ct_struct, 00h
        push    08c4h
        push    offset ct_struct
        call    returnsj
        add     sp, 04h
        push    00h
        call    exit
        add     sp, 02h
        leave
        pop     di
        pop     si
        ret
        public init_disp
init_disp:

        linkage

L10001:
        dw      0180h

        code

        linkage
```

```
L10002:
        dw      0200h

        code

        push    si
        push    di
        enter   0218h, 00h
        mov     word ptr [bp-0206h], 00h
        fldi    cs:word ptr L10001
        fstpf   word ptr [bp-0212h]
        fldi    cs:word ptr L10002
        fstpf   word ptr [bp-0216h]
        call    InitDma
        call    ginitgks
        fldf    word ptr [bp-0212h]
        fstpf   word ptr [bp-020eh]
        fldf    word ptr [bp-0216h]
        fstpf   word ptr [bp-020ah]
        lea     ax, word ptr [bp-0218h]
        push    ax
        push    03h
        lea     ax, word ptr [bp-020eh]
        push    ax
        call    StartTrans
        add     sp, 06h
        cmp     word ptr [bp-0218h], 00h
        jne     word ptr L26
        lea     ax, word ptr [bp-0218h]
        push    ax
        push    01h
        call    TransFull
        add     sp, 04h
        cmp     word ptr [bp-0218h], 00h
        jne     word ptr L26
        call    StartMount
        lea     ax, word ptr [bp-0206h]
        push    ax
        push    00h
        push    00h
        push    00h
        push    00h
        push    00h
        push    03h
```

```
        push    0ah
        push    0143h
        call    gksif
        add     sp, 012h
        lea     ax, word ptr [bp-0218h]
        push    ax
        push    013h
        push    03h
        push    06fh
        call    gksif
        add     sp, 08h
L26:
        cmp     word ptr [bp-0218h], 00h
        je      word ptr L28
        push    00h
        push    00h
        push    00h
        push    word ptr [bp-0218h]
        call    handle_exc
        add     sp, 08h
L28:
        mov     word ptr [bp-0202h], 00h
L31:
        cmp     word ptr [bp-0202h], 0100h
        jge     word ptr L29
        mov     di, word ptr [bp-0202h]
        sal     di, 01h
        add     di, bp
        mov     ax, word ptr [bp-0202h]
        mov     word ptr [di-0200h], ax
        mov     word ptr [bp-0204h], 00h
L34:
        cmp     word ptr [bp-0204h], 04h
        jge     word ptr L32
        mov     si, word ptr [bp-0202h]
        sal     si, 01h
        mov     di, word ptr [bp-0204h]
        sal     di, 09h
        add     di, si
        mov     ax, word ptr [bp-0202h]
        mov     word ptr [di+grad8bit], ax
        inc     word ptr [bp-0204h]
        jmp     word ptr L34
L32:
        inc     word ptr [bp-0202h]
```

```
             jmp     word ptr L31
L29:
             lea     ax, word ptr [bp-0218h]
             push    ax
             push    00h
             lea     ax, word ptr [bp-0200h]
             push    ax
             push    00h
             push    00h
             push    00h
             push    09h
             push    0145h
             call    gksif
             add     sp, 010h
             lea     ax, word ptr [bp-0218h]
             push    ax
             push    00h
             lea     ax, word ptr [bp-0200h]
             push    ax
             push    01h
             push    00h
             push    00h
             push    09h
             push    0145h
             call    gksif
             add     sp, 010h
             lea     ax, word ptr [bp-0218h]
             push    ax
             push    00h
             lea     ax, word ptr [bp-0200h]
             push    ax
             push    02h
             push    00h
             push    00h
             push    09h
             push    0145h
             call    gksif
             add     sp, 010h
             lea     ax, word ptr [bp-0218h]
             push    ax
             push    00h
             lea     ax, word ptr [bp-0200h]
             push    ax
             push    03h
             push    00h
```

```
        push    00h
        push    09h
        push    0145h
        call    gksif
        add     sp, 010h
        fwait
        leave
        pop     di
        pop     si
        ret
        public end_disp
end_disp:
        push    si
        push    di
        push    bp
        mov     bp, sp
        call    EndDma
        leave
        pop     di
        pop     si
        ret
        public set_smart_job_active
set_smart_job_active:
        push    si
        push    di
        push    bp
        mov     bp, sp
        leave
        pop     di
        pop     si
        ret
        public load_smart
load_smart:

        strings

L41:
        db      073h
        db      06dh
        db      061h
        db      072h
        db      074h
        db      079h
        db      00h
```

```
        code

        strings
L43:
        db      073h
        db      06dh
        db      061h
        db      072h
        db      074h
        db      00h

        code

        push    si
        push    di
        enter   04h, 00h
        push    si
        push    di
        push    es
        push    03h
        lea     ax, word ptr [bp-02h]
        push    ax
        call    rqgettasktokens
        pop     es
        pop     di
        pop     si
        mov     word ptr root_token, ax
        push    si
        push    di
        push    es
        call    prisma_token
        push    ax
        push    offset sclsem
        push    -01h
        lea     ax, word ptr [bp-02h]
        push    ax
        call    rqlookupobject
        pop     es
        pop     di
        pop     si
        mov     word ptr scanner1_sem, ax
        push    si
        push    di
```

```
push    es
call    prisma_token
push    ax
push    offset sc2sem
push    -01h
lea     ax, word ptr [bp-02h]
push    ax
call    rqlookupobject
pop     es
pop     di
pop     si
mov     word ptr scanner2_sem, ax
push    si
push    di
push    es
call    prisma_token
push    ax
push    offset sc1rdy
push    -01h
lea     ax, word ptr [bp-02h]
push    ax
call    rqlookupobject
pop     es
pop     di
pop     si
mov     word ptr scanner1_rdy, ax
push    si
push    di
push    es
call    prisma_token
push    ax
push    offset sc2rdy
push    -01h
lea     ax, word ptr [bp-02h]
push    ax
call    rqlookupobject
pop     es
pop     di
pop     si
mov     word ptr scanner2_rdy, ax
push    00h
push    word ptr scanner1_rdy
call    buildptr
add     sp, 04h
mov     word ptr sc1_rdy_ptr, ax
```

```
push    00h
push    word ptr scanner2_rdy
call    buildptr
add     sp, 04h
mov     word ptr sc2_rdy_ptr, ax
push    si
push    di
push    es
push    0eh
lea     ax, word ptr [bp-02h]
push    ax
call    rqcreatesegment
pop     es
pop     di
pop     si
mov     word ptr scan_token, ax
push    00h
push    ax
call    buildptr
add     sp, 04h
mov     word ptr scan_tok, ax
push    si
push    di
push    es
push    00h
lea     ax, word ptr [bp-02h]
push    ax
call    rqcreatemailbox
pop     es
pop     di
pop     si
mov     di, word ptr scan_tok
mov     word ptr [di+02h], ax
push    si
push    di
push    es
push    00h
lea     ax, word ptr [bp-02h]
push    ax
call    rqcreatemailbox
pop     es
pop     di
pop     si
mov     di, word ptr scan_tok
mov     word ptr [di+04h], ax
```

```
push    si
push    di
push    es
push    00h
lea     ax, word ptr [bp-02h]
push    ax
call    rqcreatemailbox
pop     es
pop     di
pop     si
mov     di, word ptr scan_tok
mov     word ptr [di+06h], ax
push    si
push    di
push    es
push    00h
lea     ax, word ptr [bp-02h]
push    ax
call    rqcreatemailbox
pop     es
pop     di
pop     si
mov     di, word ptr scan_tok
mov     word ptr [di+08h], ax
push    si
push    di
push    es
push    020h
lea     ax, word ptr [bp-02h]
push    ax
call    rqcreatesegment
pop     es
pop     di
pop     si
mov     di, word ptr scan_tok
mov     word ptr [di+0ah], ax
push    si
push    di
push    es
push    010h
lea     ax, word ptr [bp-02h]
push    ax
call    rqcreatesegment
pop     es
pop     di
```

```
pop     si
mov     di, word ptr scan_tok
mov     word ptr [di+0ch], ax
push    si
push    di
push    es
push    02h
lea     ax, word ptr [bp-02h]
push    ax
call    rqcreatesegment
pop     es
pop     di
pop     si
mov     word ptr scid_token, ax
push    00h
mov     di, word ptr scan_tok
push    word ptr [di+0ch]
call    buildptr
add     sp, 04h
mov     word ptr resp_pointer, ax
push    00h
mov     di, word ptr scan_tok
push    word ptr [di+0ah]
call    buildptr
add     sp, 04h
mov     word ptr appl_pointer, ax
push    00h
push    word ptr scid_token
call    buildptr
add     sp, 04h
mov     word ptr scid_pointer, ax
mov     di, word ptr resp_pointer
mov     word ptr [di+02h], 00h
push    si
push    di
push    es
push    04h
lea     ax, word ptr [bp-02h]
push    ax
call    rqcreatesegment
pop     es
pop     di
pop     si
mov     di, word ptr scan_tok
mov     word ptr [di], ax
```

```
        cmp     word ptr scanner_id, 01h
        jne     word ptr L38
        mov     ax, word ptr scanner1_sem
        mov     word ptr sc_curnt_sem, ax
        mov     ax, word ptr scanner2_sem
        mov     word ptr sc_other_sem, ax
        mov     ax, word ptr sc1_rdy_ptr
        mov     word ptr sc_rdy_ptr, ax
        mov     di, word ptr sc1_rdy_ptr
        mov     word ptr [di], 00h
        mov     di, word ptr scid_pointer
        mov     word ptr [di], 01h
        push    si
        push    di
        push    es
        push    word ptr root_token
        push    word ptr scan_token
        push    offset sc1tok
        jmp     word ptr L20007
L38:
        mov     ax, word ptr scanner2_sem
        mov     word ptr sc_curnt_sem, ax
        mov     ax, word ptr scanner1_sem
        mov     word ptr sc_other_sem, ax
        mov     ax, word ptr sc2_rdy_ptr
        mov     word ptr sc_rdy_ptr, ax
        mov     di, word ptr sc2_rdy_ptr
        mov     word ptr [di], 00h
        mov     di, word ptr scid_pointer
        mov     word ptr [di], 02h
        push    si
        push    di
        push    es
        push    word ptr root_token
        push    word ptr scan_token
        push    offset sc2tok
L20007:
        lea     ax, word ptr [bp-02h]
        push    ax
        call    rqcatalogobject
        pop     es
        pop     di
        pop     si
        push    si
        push    di
```

```
              push    es
              push    word ptr root_token
              push    word ptr scid_token
              push    offset scidtok
              lea     ax, word ptr [bp-04h]
              push    ax
              call    rqcatalogobject
              pop     es
              pop     di
              pop     si
              push    word ptr scanner_id
              call    init_job_communication
              add     sp, 02h
              call    smarty_active
              or      ax, ax
              je      word ptr L40
              push    00h
              push    00h
              push    02h
              push    0ah
              push    01h
              push    offset L41
              jmp     word ptr L20009
L40:
              push    00h
              push    00h
              push    02h
              push    0ah
              push    01h
              push    offset L43
L20009:
              call    loadjob
              add     sp, 0ch
              push    word ptr scanner_id
              call    put_scanner_id
              add     sp, 02h
              push    01h
              call    init_scanner
              add     sp, 02h
              leave
              pop     di
              pop     si
              ret
              public unload_smart
unload_smart:
```

```
        push    si
        push    di
        enter   02h, 00h
        cmp     word ptr scanner_id, 01h
        jne     word ptr L45
        push    si
        push    di
        push    es
        push    word ptr root_token
        push    offset sc1tok
        jmp     word ptr L20015
L45:
        push    si
        push    di
        push    es
        push    word ptr root_token
        push    offset sc2tok
L20015:
        lea     ax, word ptr [bp-02h]
        push    ax
        call    rquncatalogobject
        pop     es
        pop     di
        pop     si
        push    01h
        call    dts_finish_host
        add     sp, 02h
        mov     di, word ptr sc_rdy_ptr
        mov     word ptr [di], 01h
        push    si
        push    di
        push    es
        mov     di, word ptr scan_tok
        push    word ptr [di+02h]
        lea     ax, word ptr [bp-02h]
        push    ax
        call    rqdeletemailbox
        pop     es
        pop     di
        pop     si
        push    si
        push    di
        push    es
        mov     di, word ptr scan_tok
        push    word ptr [di+04h]
```

```
lea     ax, word ptr [bp-02h]
push    ax
call    rqdeletemailbox
pop     es
pop     di
pop     si
push    si
push    di
push    es
mov     di, word ptr scan_tok
push    word ptr [di+06h]
lea     ax, word ptr [bp-02h]
push    ax
call    rqdeletemailbox
pop     es
pop     di
pop     si
push    si
push    di
push    es
mov     di, word ptr scan_tok
push    word ptr [di+08h]
lea     ax, word ptr [bp-02h]
push    ax
call    rqdeletemailbox
pop     es
pop     di
pop     si
push    si
push    di
push    es
mov     di, word ptr scan_tok
push    word ptr [di+0ah]
lea     ax, word ptr [bp-02h]
push    ax
call    rqdeletesegment
pop     es
pop     di
pop     si
push    si
push    di
push    es
mov     di, word ptr scan_tok
push    word ptr [di+0ch]
lea     ax, word ptr [bp-02h]
```

```
        push    ax
        call    rqdeletesegment
        pop     es
        pop     di
        pop     si
        push    si
        push    di
        push    es
        mov     di, word ptr scan_tok
        push    word ptr [di]
        lea     ax, word ptr [bp-02h]
        push    ax
        call    rqdeletesegment
        pop     es
        pop     di
        pop     si
        leave
        pop     di
        pop     si
        ret
        public decode_rec
decode_rec:

        strings

L50:
        db      06ch
        db      06fh
        db      061h
        db      064h
        db      020h
        db      044h
        db      045h
        db      043h
        db      04fh
        db      044h
        db      045h
        db      020h
        db      06ah
        db      06fh
        db      062h
        db      0ah
        db      00h

        code
```

```
        strings
L51:
        db      03ah
        db      073h
        db      064h
        db      03ah
        db      070h
        db      072h
        db      069h
        db      073h
        db      06dh
        db      061h
        db      02fh
        db      073h
        db      079h
        db      073h
        db      074h
        db      065h
        db      06dh
        db      02fh
        db      064h
        db      065h
        db      063h
        db      06fh
        db      064h
        db      065h
        db      00h

        code

        push    si
        push    di
        push    bp
        mov     bp, sp
        cmpb    byte ptr glpar+023h, 00h
        je      word ptr L49
        push    offset L50
        call    printf
        add     sp, 02h
L49:
        push    offset L51
        call    unix_system
```

```
        add      sp, 02h
        leave
        pop      di
        pop      si
        ret
        public reset_control
reset_control:
        push     si
        push     di
        push     bp
        mov      bp, sp
        leave
        pop      di
        pop      si
        ret
        public stretch_intgrad
stretch_intgrad:
        push     si
        push     di
        push     bp
        mov      bp, sp
        leave
        pop      di
        pop      si
        ret
        public set_vip2_or_bip
set_vip2_or_bip:
        push     si
        push     di
        push     bp
        mov      bp, sp
        leave
        pop      di
        pop      si
        ret
        public lut_correct_grad
lut_correct_grad:
        push     si
        push     di
        push     bp
        mov      bp, sp
        leave
        pop      di
        pop      si
        ret
```

```
        public update_tan
update_tan:
        push    si
        push    di
        push    bp
        mov     bp, sp
        leave
        pop     di
        pop     si
        ret
```

APPENXIX C

-SRV2$DUA1:[USERS.SMART.KEIDAR_MJK.RGB;3

[O R I G I N A L]

	R	G	B
1	0	0	0
2	0	0	0
3	3	3	0
4	3	0	3
5	0	3	3
6	2	2	2
7	5	6	7
8	12	2	2
9	2	12	2
10	12	12	2
11	2	2	12
12	12	2	12
13	2	12	12
14	12	12	12
15	17	3	3
16	3	17	3
17	3	3	17
18	16	16	16
19	27	26	25
20	34	3	3
21	37	6	7
22	51	3	3
23	48	16	16
24	59	26	25
25	3	34	3
26	5	38	7
27	3	51	3
28	16	48	16
29	27	58	25
30	3	3	34
31	5	6	39
32	3	3	51
33	16	16	48
34	27	26	57
35	5	5	5
36	5	5	5
37	3	3	68
38	5	6	71
39	3	3	85
40	16	16	80
41	27	26	89
42	5	38	39
43	16	48	48
44	27	58	57
45	3	68	3
46	5	70	7
47	3	85	3
48	16	80	16
49	27	90	25
50	37	38	7
51	48	48	16
52	59	58	25
53	37	6	39
54	48	16	48
55	59	26	57
56	68	3	3

57	69	6	7

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJK.RGB;3

58	85	3	3
59	80	16	16
60	91	26	25
61	102	3	3
62	101	6	7
63	119	3	3
64	112	16	16
65	123	26	25
66	69	38	7
67	80	48	16
68	91	58	25
69	10	10	10
70	10	10	10
71	69	6	39
72	80	16	48
73	91	26	57
74	37	6	71
75	48	16	80
76	59	26	89
77	37	38	39
78	59	58	57
79	37	70	7
80	48	80	16
81	59	90	25
82	3	102	3
83	5	102	7
84	3	119	3
85	16	112	16
86	27	122	25
87	5	70	39
88	16	80	48
89	27	90	57
90	5	38	71
91	16	48	80
92	27	58	89
93	3	3	102
94	5	6	103
95	3	3	119
96	16	16	112
97	27	26	121
98	5	6	135
99	3	3	136
100	3	3	153
101	16	16	144
102	27	26	153
103	16	16	16
104	16	16	16
105	5	38	103
106	16	48	112
107	27	58	121
108	5	70	71
109	16	80	80
110	27	90	89
111	5	102	39
112	16	112	48
113	27	122	57
114	5	134	7
115	3	136	3
116	3	153	3
117	16	144	16
118	27	154	25

119 37 102 7

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJK.RGB;3

120	48	112	16
121	59	122	25
122	37	70	39
123	59	90	57
124	37	38	71
125	59	58	89
126	37	6	103
127	48	16	112
128	59	26	121
129	69	6	71
130	80	16	80
131	91	26	89
132	69	38	39
133	91	58	57
134	69	70	7
135	80	80	16
136	91	90	25
137	21	21	21
138	21	21	21
139	101	38	7
140	112	48	16
141	123	58	25
142	101	6	39
143	112	16	48
144	123	26	57
145	133	6	7
146	136	3	3
147	153	3	3
148	144	16	16
149	155	26	25
150	165	6	7
151	170	3	3
152	187	3	3
153	176	16	16
154	187	26	25
155	133	38	7
156	144	48	16
157	155	58	25
158	133	6	39
159	144	16	48
160	155	26	57
161	101	6	71
162	112	16	80
163	123	26	89
164	101	38	39
165	123	58	57
166	101	70	7
167	112	80	16
168	120	91	24
169	69	102	7
170	80	112	16
171	26	26	26
172	26	26	26
173	91	122	25
174	69	70	39
175	91	90	57
176	69	38	71
177	91	58	89
178	69	6	103
179	80	16	112
180	91	26	121

181	37	6	135

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJK.RGB;3

182	48	16	144
183	59	26	153
184	37	38	103
185	59	58	121
186	37	70	71
187	59	90	89
188	37	102	39
189	59	122	57
190	37	134	7
191	48	144	16
192	59	154	25
193	5	166	7
194	3	170	3
195	3	187	3
196	16	176	16
197	27	186	25
198	5	134	39
199	16	144	48
200	27	154	57
201	5	102	71
202	17	113	81
203	27	122	89
204	5	70	103
205	31	31	31
206	31	31	31
207	16	80	112
208	27	90	121
209	5	38	135
210	16	48	144
211	27	58	153
212	5	6	167
213	3	3	170
214	3	3	187
215	16	16	176
216	27	26	185
217	5	6	199
218	3	3	204
219	3	3	221
220	16	16	208
221	27	26	217
222	5	38	167
223	16	48	176
224	27	58	185
225	5	70	135
226	16	80	144
227	27	90	153
228	5	102	103
229	16	112	112
230	27	122	121
231	5	134	71
232	16	144	80
233	27	154	89
234	5	166	39
235	16	176	48
236	26	187	54
237	5	198	7
238	3	204	3
239	36	36	36
240	36	36	36
241	3	221	3
242	16	208	16

243	27	218	25

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJK.RGB;3

244	37	166	7
245	48	176	16
246	59	186	25
247	37	134	39
248	59	154	57
249	37	102	71
250	59	122	89
251	37	70	103
252	59	90	121
253	37	38	135
254	59	58	153
255	37	6	167
256	48	16	176
257	59	26	185
258	69	6	135
259	80	16	144
260	91	26	153
261	69	38	103
262	91	58	121
263	69	70	71
264	91	90	89
265	69	102	39
266	91	122	57
267	69	134	7
268	80	144	16
269	91	154	25
270	102	103	8
271	112	112	16
272	123	122	25
273	42	42	42
274	42	42	42
275	101	70	39
276	123	90	57
277	101	38	71
278	123	58	89
279	101	6	103
280	112	16	112
281	123	26	121
282	133	6	71
283	144	16	80
284	155	26	89
285	133	38	39
286	155	58	57
287	133	70	7
288	144	80	16
289	155	90	25
290	165	38	7
291	176	48	16
292	187	58	25
293	165	6	39
294	176	16	48
295	187	26	57
296	197	6	7
297	204	3	3
298	221	3	3
299	208	16	16
300	219	26	25
301	229	6	7
302	238	3	3
303	241	2	2
304	251	2	2

305	253	3	0

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJK.RGB;3

306	253	0	3
307	47	47	47
308	47	47	47
309	254	3	3
310	241	12	2
311	251	12	2
312	241	2	12
313	251	2	12
314	241	12	12
315	251	12	12
316	253	17	3
317	253	3	17
318	240	16	16
319	251	26	25
320	197	38	7
321	208	48	16
322	219	58	25
323	197	6	39
324	208	16	48
325	219	26	57
326	165	6	71
327	176	16	80
328	187	26	89
329	165	38	39
330	187	58	57
331	165	70	7
332	176	80	16
333	187	90	25
334	133	102	7
335	144	112	16
336	155	122	25
337	133	70	39
338	154	87	58
339	133	38	71
340	155	58	89
341	52	52	52
342	52	52	52
343	133	6	103
344	144	16	112
345	155	26	121
346	101	6	135
347	112	16	144
348	123	26	153
349	101	38	103
350	123	58	121
351	101	70	71
352	123	90	89
353	101	102	39
354	123	122	57
355	101	134	7
356	112	144	16
357	123	154	25
358	69	166	7
359	80	176	16
360	91	186	25
361	69	134	39
362	91	154	57
363	69	102	71
364	91	122	89
365	69	70	103
366	91	90	121

367 69 38 135

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJK.RGB;3

368	91	58	153
369	69	6	167
370	80	16	176
371	91	26	185
372	38	7	200
373	48	16	208
374	59	26	217
375	57	57	57
376	57	57	57
377	37	38	167
378	59	58	185
379	37	70	135
380	59	90	153
381	37	102	103
382	59	122	121
383	37	134	71
384	59	154	89
385	37	166	39
386	59	186	57
387	37	198	7
388	48	208	16
389	59	218	25
390	5	230	7
391	3	238	3
392	2	241	2
393	12	241	2
394	2	251	2
395	3	253	0
396	0	253	3
397	3	254	3
398	12	251	2
399	2	241	12
400	12	241	12
401	2	251	12
402	12	251	12
403	17	253	3
404	3	253	17
405	16	240	16
406	26	247	26
407	5	198	39
408	16	208	48
409	62	62	62
410	62	62	62
411	27	218	57
412	5	166	71
413	16	176	80
414	27	186	89
415	5	134	103
416	16	144	112
417	27	154	121
418	5	102	135
419	16	112	144
420	27	122	153
421	5	70	167
422	16	80	176
423	27	90	185
424	5	38	199
425	16	48	208
426	27	58	217
427	5	6	231
428	3	3	238

429	2	2	241

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJK.RGB;3

430	12	2	241
431	2	12	241
432	12	12	241
433	2	2	251
434	3	0	253
435	0	3	253
436	3	3	254
437	12	2	251
438	2	12	251
439	12	12	251
440	16	4	253
441	3	17	253
442	16	16	240
443	68	68	68
444	68	68	68
445	27	26	249
446	5	38	231
447	3	34	253
448	3	51	253
449	16	48	240
450	27	58	249
451	5	70	199
452	16	80	208
453	27	90	217
454	5	102	167
455	16	112	176
456	27	122	185
457	5	134	135
458	16	144	144
459	27	154	153
460	5	166	103
461	16	176	112
462	27	186	121
463	5	198	71
464	16	208	80
465	27	218	89
466	5	230	39
467	3	253	34
468	3	253	51
469	16	240	48
470	27	250	57
471	37	230	7
472	34	253	3
473	51	253	3
474	49	241	17
475	59	250	25
476	37	198	39
477	73	73	73
478	73	73	73
479	59	218	57
480	37	166	71
481	59	186	89
482	37	134	103
483	59	154	121
484	37	102	135
485	59	122	153
486	37	70	167
487	59	90	185
488	37	38	199
489	59	58	217
490	37	6	231

491	34	3	253

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJK.RGB;3

492	51	3	253
493	48	16	240
494	59	26	249
495	69	6	199
496	80	16	208
497	91	26	217
498	69	38	167
499	91	58	185
500	69	70	135
501	91	90	153
502	69	102	103
503	91	122	121
504	69	134	71
505	91	154	89
506	69	166	39
507	91	186	57
508	70	199	8
509	80	208	16
510	91	218	25
511	78	78	78
512	78	78	78
513	101	166	7
514	112	176	16
515	123	186	25
516	101	134	39
517	123	154	57
518	101	102	71
519	123	122	89
520	101	70	103
521	123	90	121
522	101	38	135
523	123	58	153
524	101	6	167
525	112	16	176
526	123	26	185
527	133	6	135
528	144	16	144
529	155	26	153
530	133	38	103
531	155	58	121
532	133	70	71
533	155	90	89
534	133	102	39
535	155	122	57
536	133	134	7
537	144	144	16
538	155	154	25
539	165	102	7
540	176	112	16
541	187	122	25
542	166	71	40
543	187	90	57
544	165	38	71
545	83	83	83
546	83	83	83
547	187	58	89
548	165	6	103
549	176	16	112
550	187	26	121
551	197	6	71
552	208	16	80

553	219	26	89

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJK.RGB;3

554	197	38	39
555	219	58	57
556	197	70	7
557	208	80	16
558	219	90	25
559	229	38	7
560	253	34	3
561	253	51	3
562	240	48	16
563	251	58	25
564	229	6	39
565	253	3	34
566	253	3	51
567	240	16	48
568	251	26	57
569	229	6	71
570	253	3	68
571	253	3	85
572	240	16	80
573	251	26	89
574	229	38	39
575	240	48	48
576	250	59	54
577	229	70	7
578	253	68	3
579	88	88	88
580	88	88	88
581	253	85	3
582	240	80	16
583	251	90	25
584	197	102	7
585	208	112	16
586	219	122	25
587	197	70	39
588	219	90	57
589	197	38	71
590	219	58	89
591	197	6	103
592	208	16	112
593	219	26	121
594	165	6	135
595	176	16	144
596	187	26	153
597	165	38	103
598	187	58	121
599	165	70	71
600	187	90	89
601	165	102	39
602	187	122	57
603	165	134	7
604	176	144	16
605	187	154	25
606	133	166	7
607	144	176	16
608	155	186	25
609	133	134	39
610	154	151	58
611	133	102	71
612	155	122	89
613	94	94	94
614	94	94	94

615	133	70	103

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJK.RGB;3

616	155	90	121
617	133	38	135
618	155	58	153
619	133	6	167
620	144	16	176
621	155	26	185
622	101	6	199
623	112	16	208
624	123	26	217
625	101	38	167
626	123	58	185
627	101	70	135
628	123	90	153
629	101	102	103
630	123	122	121
631	101	134	71
632	123	154	89
633	101	166	39
634	123	186	57
635	101	198	7
636	112	208	16
637	123	218	25
638	69	230	7
639	68	253	3
640	85	253	3
641	80	240	16
642	91	250	25
643	69	198	39
644	90	215	58
645	69	166	71
646	91	186	89
647	99	99	99
648	99	99	99
649	69	134	103
650	91	154	121
651	69	102	135
652	91	122	153
653	69	70	167
654	91	90	185
655	69	38	199
656	91	58	217
657	69	6	231
658	68	3	253
659	85	3	253
660	80	16	240
661	91	26	249
662	37	38	231
663	48	48	240
664	59	58	249
665	37	70	199
666	59	90	217
667	37	102	167
668	59	122	185
669	37	134	135
670	59	154	153
671	37	166	103
672	59	186	121
673	37	198	71
674	59	218	89
675	37	230	39
676	48	240	48

677	59	250	57

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJK.RGB;3

678	5	231	71
679	3	253	68
680	3	253	85
681	104	104	104
682	104	104	104
683	16	240	80
684	27	250	89
685	5	198	103
686	16	208	112
687	27	218	121
688	5	166	135
689	16	176	144
690	27	186	153
691	5	134	167
692	16	144	176
693	27	154	185
694	5	102	199
695	16	112	208
696	27	122	217
697	5	70	231
698	3	68	253
699	3	85	253
700	16	80	240
701	27	90	249
702	5	102	231
703	3	102	253
704	3	119	253
705	16	112	240
706	27	122	249
707	5	134	199
708	16	144	208
709	27	154	217
710	5	166	167
711	16	176	176
712	26	187	182
713	5	198	135
714	16	208	144
715	109	109	109
716	109	109	109
717	27	218	153
718	5	230	103
719	3	253	102
720	3	253	119
721	16	240	112
722	27	250	121
723	37	230	71
724	48	240	80
725	59	250	89
726	37	198	103
727	59	218	121
728	37	166	135
729	59	186	153
730	37	134	167
731	59	154	185
732	37	102	199
733	59	122	217
734	37	70	231
735	48	80	240
736	59	90	249
737	69	38	231
738	80	48	240

739	91	58	249

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJK.RGB;

740	69	70	199
741	91	90	217
742	69	102	167
743	91	122	185
744	69	134	135
745	91	154	153
746	70	167	104
747	91	186	121
748	69	198	71
749	114	114	114
750	114	114	114
751	91	218	89
752	69	230	39
753	80	240	48
754	91	250	57
755	101	230	7
756	102	253	3
757	119	253	3
758	112	240	16
759	123	250	25
760	101	198	39
761	123	218	57
762	101	166	71
763	123	186	89
764	101	134	103
765	123	154	121
766	101	102	135
767	123	122	153
768	101	70	167
769	123	90	185
770	101	38	199
771	123	58	217
772	101	6	231
773	102	3	253
774	119	3	253
775	112	16	240
776	123	26	249
777	133	6	199
778	144	16	208
779	155	26	217
780	134	39	168
781	155	58	185
782	133	70	135
783	120	120	120
784	120	120	120
785	155	90	153
786	133	102	103
787	155	122	121
788	133	134	71
789	155	154	89
790	133	166	39
791	155	186	57
792	133	198	7
793	144	208	16
794	155	218	25
795	165	166	7
796	176	176	16
797	187	186	25
798	165	134	39
799	187	154	57
800	165	102	71

801	187	122	89

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJK.RGB;3

802	165	70	103
803	187	90	121
804	165	38	135
805	187	58	153
806	165	6	167
807	176	16	176
808	187	26	185
809	197	6	135
810	208	16	144
811	219	26	153
812	197	38	103
813	219	58	121
814	198	71	72
815	219	90	89
816	197	102	39
817	125	125	125
818	125	125	125
819	219	122	57
820	197	134	7
821	208	144	16
822	219	154	25
823	229	102	7
824	253	102	3
825	253	119	3
826	240	112	16
827	251	122	25
828	229	70	39
829	240	80	48
830	251	90	57
831	229	38	71
832	240	48	80
833	251	58	89
834	229	6	103
835	253	3	102
836	253	3	119
837	240	16	112
838	251	26	121
839	229	6	135
840	253	3	136
841	253	3	153
842	240	16	144
843	251	26	153
844	230	39	104
845	240	48	112
846	251	58	121
847	229	70	71
848	241	81	81
849	251	90	89
850	229	102	39
851	130	130	130
852	130	130	130
853	240	112	48
854	251	122	57
855	229	134	7
856	253	136	3
857	253	153	3
858	240	144	16
859	251	154	25
860	197	166	7
861	208	176	16
862	219	186	25

863	197	134	39

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJK.RGB;3

864	219	154	57
865	197	102	71
866	219	122	89
867	197	70	103
868	219	90	121
869	197	38	135
870	219	58	153
871	197	6	167
872	208	16	176
873	219	26	185
874	165	6	199
875	176	16	208
876	187	26	217
877	165	38	167
878	186	59	182
879	165	70	135
880	187	90	153
881	165	102	103
882	186	123	118
883	165	134	71
884	187	154	89
885	135	135	135
886	135	135	135
887	165	166	39
888	187	186	57
889	165	198	7
890	176	208	16
891	187	218	25
892	133	230	7
893	136	253	3
894	153	253	3
895	144	240	16
896	155	250	25
897	133	198	39
898	155	218	57
899	133	166	71
900	155	186	89
901	133	134	103
902	155	154	121
903	133	102	135
904	155	122	153
905	133	70	167
906	155	90	185
907	133	38	199
908	155	58	217
909	133	6	231
910	136	3	253
911	153	3	253
912	145	17	241
913	155	26	249
914	101	38	231
915	112	48	240
916	122	59	246
917	101	70	199
918	123	90	217
919	141	141	141
920	141	141	141
921	101	102	167
922	123	122	185
923	101	134	135
924	123	154	153

925	101	166	103

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJK.RGB;3

926	123	186	121
927	101	198	71
928	123	218	89
929	101	230	39
930	112	240	48
931	123	250	57
932	69	230	71
933	80	240	80
934	91	250	89
935	69	198	103
936	91	218	121
937	69	166	135
938	91	186	153
939	69	134	167
940	91	154	185
941	69	102	199
942	91	122	217
943	69	70	231
944	80	80	240
945	91	90	249
946	38	103	232
947	48	112	240
948	59	122	249
949	40	137	201
950	58	155	214
951	37	166	167
952	59	186	185
953	146	146	146
954	146	146	146
955	37	198	135
956	59	218	153
957	37	230	103
958	48	240	112
959	59	250	121
960	5	230	135
961	3	253	136
962	3	253	153
963	16	240	144
964	27	250	153
965	5	198	167
966	16	208	176
967	27	218	185
968	5	166	199
969	16	176	208
970	27	186	217
971	5	134	231
972	3	136	253
973	3	153	253
974	16	144	240
975	27	154	249
976	5	166	231
977	3	170	253
978	3	187	253
979	16	176	240
980	26	187	246
981	5	198	199
982	16	208	208
983	24	220	211
984	5	231	167
985	3	253	170
986	3	253	187

987	151	151	151

_SEV2$DUA1:[USERS.SMART.KEIMAR_MJK.RGB;3

988	151	151	151
989	16	240	176
990	27	250	185
991	37	230	135
992	48	240	144
993	59	250	153
994	37	198	167
995	59	218	185
996	37	166	199
997	59	186	217
998	37	134	231
999	48	144	240
1000	59	154	249
1001	69	102	231
1002	80	112	240
1003	91	122	249
1004	69	134	199
1005	91	154	217
1006	69	166	167
1007	91	186	185
1008	69	198	135
1009	91	218	153
1010	69	230	103
1011	80	240	112
1012	91	250	121
1013	101	230	71
1014	113	241	81
1015	123	250	89
1016	101	198	103
1017	120	212	123
1018	102	167	136
1019	123	186	153
1020	101	134	167
1021	156	156	156
1022	156	156	156
1023	123	154	185
1024	101	102	199
1025	123	122	217
1026	101	70	231
1027	112	80	240
1028	123	90	249
1029	133	38	231
1030	144	48	240
1031	155	58	249
1032	133	70	199
1033	155	90	217
1034	133	102	167
1035	155	122	185
1036	133	134	135
1037	155	154	153
1038	133	166	103
1039	155	186	121
1040	133	198	71
1041	155	218	89
1042	133	230	39
1043	144	240	48
1044	155	250	57
1045	165	230	7
1046	170	253	3
1047	187	253	3
1048	177	241	17

1049	187	250	25

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJK.RGB;3

1050	165	198	39
1051	184	212	59
1052	166	167	72
1053	187	186	89
1054	165	134	103
1055	161	161	161
1056	161	161	161
1057	187	154	121
1058	165	102	135
1059	187	122	153
1060	165	70	167
1061	187	90	185
1062	165	38	199
1063	187	58	217
1064	165	6	231
1065	170	3	253
1066	187	3	253
1067	176	16	240
1068	187	26	249
1069	197	6	199
1070	208	16	208
1071	219	26	217
1072	197	38	167
1073	219	58	185
1074	197	70	135
1075	219	90	153
1076	197	102	103
1077	219	122	121
1078	197	134	71
1079	219	154	89
1080	197	166	39
1081	219	186	57
1082	198	199	8
1083	208	208	16
1084	219	218	25
1085	232	167	7
1086	253	171	3
1087	253	187	3
1088	214	162	32
1089	167	167	167
1090	167	167	167
1091	251	186	25
1092	229	134	39
1093	240	144	48
1094	251	154	57
1095	229	102	71
1096	240	112	80
1097	251	122	89
1098	229	70	103
1099	240	80	112
1100	251	90	121
1101	229	38	135
1102	240	48	144
1103	251	58	153
1104	229	6	167
1105	253	3	170
1106	253	3	187
1107	240	16	176
1108	251	26	185
1109	229	6	199
1110	253	3	204

1111	253	3	221

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJK.RGB;3

1112	240	16	208
1113	251	26	217
1114	229	38	167
1115	240	48	176
1116	250	59	182
1117	229	70	135
1118	240	80	144
1119	248	92	147
1120	230	103	104
1121	240	112	112
1122	223	118	117
1123	172	172	172
1124	172	172	172
1125	229	134	71
1126	240	144	80
1127	251	154	89
1128	229	166	39
1129	240	176	48
1130	251	186	57
1131	229	198	7
1132	253	204	3
1133	253	221	3
1134	240	208	16
1135	251	218	25
1136	197	230	7
1137	204	253	3
1138	221	253	3
1139	208	240	16
1140	219	250	25
1141	197	198	39
1142	219	218	57
1143	197	166	71
1144	219	186	89
1145	197	134	103
1146	219	154	121
1147	197	102	135
1148	219	122	153
1149	197	70	167
1150	218	87	186
1151	197	38	199
1152	219	58	217
1153	198	6	234
1154	205	3	253
1155	221	3	253
1156	188	32	214
1157	177	177	177
1158	177	177	177
1159	219	26	249
1160	165	38	231
1161	176	48	240
1162	187	58	249
1163	165	70	199
1164	187	90	217
1165	165	102	167
1166	187	122	185
1167	165	134	135
1168	187	154	153
1169	165	166	103
1170	187	186	121
1171	165	198	71
1172	187	218	89

1173	165	230	39

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJK.RGB;3

1174	176	240	48
1175	187	250	57
1176	133	230	71
1177	144	240	80
1178	155	250	89
1179	133	198	103
1180	155	218	121
1181	133	166	135
1182	155	186	153
1183	133	134	167
1184	154	151	186
1185	133	102	199
1186	155	122	217
1187	134	71	232
1188	145	81	241
1189	155	90	249
1190	101	102	206
1191	182	182	182
1192	182	182	182
1193	112	112	240
1194	123	122	249
1195	101	134	199
1196	123	154	217
1197	101	166	167
1198	123	186	185
1199	101	198	135
1200	123	218	153
1201	101	230	103
1202	112	240	112
1203	123	250	121
1204	69	230	135
1205	80	240	144
1206	91	250	153
1207	69	198	167
1208	91	218	185
1209	69	166	199
1210	91	186	217
1211	69	134	231
1212	80	144	240
1213	91	154	249
1214	37	166	231
1215	48	176	240
1216	59	186	249
1217	37	198	199
1218	58	219	214
1219	37	230	167
1220	48	240	176
1221	52	248	187
1222	5	231	199
1223	3	253	204
1224	21	224	198
1225	187	187	187
1226	187	187	187
1227	16	240	208
1228	27	250	217
1229	5	198	231
1230	3	204	253
1231	3	221	253
1232	16	208	240
1233	27	218	249
1234	5	230	231

1235	3	[illegible]	238

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJK.RGB;3

1236	3	238	253
1237	2	241	241
1238	12	241	241
1239	2	251	241
1240	12	251	241
1241	2	241	251
1242	12	241	251
1243	2	251	251
1244	0	253	253
1245	3	254	253
1246	3	253	254
1247	12	251	251
1248	16	240	240
1249	17	253	253
1250	27	250	249
1251	37	230	199
1252	49	241	209
1253	59	250	217
1254	37	198	231
1255	49	209	241
1256	60	215	248
1257	69	166	231
1258	84	162	214
1259	193	193	193
1260	193	193	193
1261	91	186	249
1262	69	198	199
1263	91	218	217
1264	69	230	167
1265	80	240	176
1266	91	250	185
1267	101	230	135
1268	112	240	144
1269	123	250	153
1270	101	178	167
1271	123	218	185
1272	101	166	199
1273	123	186	217
1274	101	134	231
1275	112	144	240
1276	123	154	249
1277	133	102	231
1278	144	112	240
1279	155	122	249
1280	133	134	199
1281	155	154	217
1282	133	166	167
1283	155	186	185
1284	133	198	135
1285	155	218	153
1286	134	231	104
1287	144	240	112
1288	155	250	121
1289	166	231	72
1290	177	241	81
1291	187	250	89
1292	153	180	102
1293	198	198	198
1294	198	193	198
1295	187	218	121
1296	165	166	135

1297	187	186	153

SRV2$DUA1:[USERS.SMART.KEIDAR_HJK.RGB;3

1298	165	134	167
1299	187	154	185
1300	165	102	199
1301	187	122	217
1302	165	70	231
1303	176	80	240
1304	187	90	249
1305	197	38	231
1306	208	48	240
1307	219	58	249
1308	197	70	199
1309	219	90	217
1310	197	102	167
1311	219	122	185
1312	197	134	135
1313	219	154	153
1314	197	166	103
1315	219	186	121
1316	197	198	71
1317	219	218	89
1318	197	230	39
1319	208	240	48
1320	220	249	54
1321	229	230	7
1322	253	238	3
1323	238	252	3
1324	242	241	2
1325	251	241	2
1326	215	223	20
1327	203	203	203
1328	203	203	203
1329	251	251	2
1330	253	253	0
1331	254	253	3
1332	253	254	3
1333	241	241	12
1334	251	241	12
1335	241	251	12
1336	251	251	12
1337	240	240	16
1338	253	253	17
1339	251	250	25
1340	229	198	39
1341	240	208	48
1342	251	218	57
1343	229	166	71
1344	240	176	80
1345	251	186	89
1346	229	134	103
1347	240	144	112
1348	251	154	121
1349	229	102	135
1350	240	112	144
1351	251	122	153
1352	229	70	167
1353	240	80	176
1354	250	87	186
1355	229	38	199
1356	240	48	208
1357	248	52	219
1358	231	6	231

1337	253	3	238

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJK.RGB;3

1360	212	21	224
1361	208	208	208
1362	208	208	208
1363	241	2	241
1364	251	2	241
1365	241	12	241
1366	251	12	241
1367	241	2	251
1368	251	2	251
1369	253	0	253
1370	254	3	253
1371	253	3	254
1372	241	12	251
1373	251	12	251
1374	240	16	240
1375	253	17	253
1376	251	26	249
1377	229	38	231
1378	253	34	253
1379	240	48	240
1380	253	51	253
1381	251	58	249
1382	229	70	199
1383	240	80	208
1384	251	90	217
1385	229	102	167
1386	240	112	176
1387	251	122	185
1388	230	135	136
1389	240	144	144
1390	251	154	153
1391	230	167	104
1392	241	177	113
1393	251	186	121
1394	205	180	76
1395	213	213	213
1396	213	213	213
1397	240	208	80
1398	251	218	89
1399	229	230	39
1400	253	253	34
1401	240	240	48
1402	253	253	51
1403	251	250	57
1404	197	230	71
1405	208	240	80
1406	219	250	89
1407	197	198	103
1408	219	218	121
1409	197	166	135
1410	219	186	153
1411	197	134	167
1412	219	154	185
1413	197	102	199
1414	219	122	217
1415	197	70	231
1416	208	80	240
1417	219	90	249
1418	165	102	231
1419	176	112	240
1420	187	122	249

1421	165	134	199

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJK.RGB;3

1422	186	155	214
1423	165	166	167
1424	187	186	185
1425	168	201	137
1426	186	219	150
1427	165	230	103
1428	162	214	110
1429	219	219	219
1430	219	219	219
1431	187	250	121
1432	133	230	135
1433	144	240	144
1434	155	250	153
1435	133	198	167
1436	155	218	185
1437	133	166	199
1438	155	186	217
1439	133	134	231
1440	144	144	240
1441	155	154	249
1442	101	166	231
1443	112	176	240
1444	123	186	249
1445	101	198	199
1446	123	218	217
1447	101	230	167
1448	112	240	176
1449	123	250	185
1450	69	230	199
1451	80	240	208
1452	91	250	217
1453	69	198	231
1454	80	208	240
1455	91	218	249
1456	37	231	232
1457	34	253	253
1458	48	240	240
1459	52	253	253
1460	60	249	248
1461	69	230	231
1462	74	224	24
1463	224	224	224
1464	224	224	224
1465	80	240	240
1466	85	253	253
1467	91	250	249
1468	101	230	199
1469	112	240	208
1470	123	250	217
1471	101	198	231
1472	112	208	240
1473	123	218	249
1474	133	166	231
1475	144	176	240
1476	155	186	249
1477	133	198	199
1478	155	218	217
1479	133	230	167
1480	144	240	176
1481	155	250	185
1482	165	230	135

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJK.RGB;3

1484	187	250	153
1485	165	198	167
1486	186	215	186
1487	165	166	199
1488	187	186	217
1489	165	134	231
1490	177	145	241
1491	187	154	249
1492	197	102	231
1493	209	113	241
1494	218	123	246
1495	197	134	199
1496	197	144	195
1497	229	229	229
1498	229	229	229
1499	197	166	167
1500	219	186	185
1501	197	198	135
1502	219	218	153
1503	197	230	103
1504	208	240	112
1505	219	250	121
1506	229	230	71
1507	253	253	68
1508	240	240	80
1509	253	253	85
1510	251	250	89
1511	229	198	103
1512	240	208	112
1513	251	218	121
1514	229	166	135
1515	240	176	144
1516	251	186	153
1517	229	134	167
1518	240	144	176
1519	251	154	185
1520	230	103	200
1521	240	112	208
1522	251	122	217
1523	229	70	231
1524	252	69	252
1525	240	80	240
1526	253	85	253
1527	248	92	247
1528	231	102	232
1529	253	102	253
1530	214	110	214
1531	234	234	234
1532	234	234	234
1533	253	119	253
1534	251	122	249
1535	229	134	199
1536	240	144	208
1537	251	154	217
1538	229	166	167
1539	240	176	176
1540	251	186	185
1541	229	198	135
1542	240	208	144
1543	251	218	153
1544	229	230	103

1545	253	253	102

_SKV2$DUA1:[USERS.SMART.KEIDAR_MJK.RGB;3

1546	240	240	112
1547	253	253	119
1548	251	250	121
1549	197	230	135
1550	208	240	144
1551	219	238	153
1552	197	198	167
1553	219	218	185
1554	198	167	200
1555	219	186	217
1556	197	134	231
1557	208	144	240
1558	216	155	248
1559	165	166	231
1560	176	176	240
1561	184	188	243
1562	166	199	200
1563	187	218	217
1564	153	206	154
1565	239	239	239
1566	239	239	239
1567	176	240	176
1568	187	250	185
1569	133	230	199
1570	144	240	208
1571	155	250	217
1572	133	198	231
1573	144	208	240
1574	155	218	249
1575	101	230	231
1576	102	253	253
1577	112	240	240
1578	119	253	253
1579	123	250	249
1580	133	230	231
1581	136	253	253
1582	144	240	240
1583	155	250	249
1584	153	253	253
1585	165	230	199
1586	176	240	208
1587	187	250	217
1588	166	199	232
1589	176	208	240
1590	187	218	249
1591	197	166	231
1592	209	177	241
1593	219	186	249
1594	197	198	199
1595	216	220	211
1596	198	231	168
1597	208	240	176
1598	197	222	169
1599	245	245	245
1600	245	245	245
1601	229	230	135
1602	253	253	136
1603	240	240	144
1604	251	250	153
1605	253	253	153
1606	229	198	167

1607	240	208	176

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJK.RGB;3

1608	251	218	185
1609	229	166	199
1610	240	176	208
1611	251	186	217
1612	229	134	231
1613	253	136	253
1614	240	144	240
1615	251	154	249
1616	253	153	253
1617	229	166	231
1618	253	170	253
1619	240	176	240
1620	251	186	249
1621	253	187	253
1622	230	199	200
1623	240	208	208
1624	251	218	217
1625	232	233	167
1626	252	252	170
1627	240	240	176
1628	251	250	185
1629	246	250	189
1630	198	231	200
1631	208	240	208
1632	197	222	195
1633	250	250	250
1634	250	250	250
1635	197	198	231
1636	208	208	240
1637	219	218	249
1638	165	230	231
1639	170	253	253
1640	176	240	240
1641	187	250	249
1642	187	253	253
1643	197	230	231
1644	204	253	253
1645	208	240	240
1646	219	250	249
1647	221	253	253
1648	229	230	199
1649	253	253	204
1650	240	240	208
1651	251	250	217
1652	253	253	221
1653	229	198	231
1654	253	204	253
1655	240	208	240
1656	251	218	249
1657	253	221	253
1658	229	230	231
1659	253	251	240
1660	252	239	253
1661	238	253	253
1662	240	240	240
1663	242	241	241
1664	250	242	241
1665	241	251	241
1666	223	223	215
1667	254	254	254
1668	254	254	254
1669	241	241	251

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJK.RGB;3

1670	251	241	251
1671	241	251	251
1672	251	250	249
1673	251	251	251
1674	254	253	253
1675	253	254	253
1676	253	253	254

==

APPENDIX D

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJDB.RGB;3

==

	R	G	B	[4 C A S T]
1	75	66	53	
2	74	65	50	
3	77	66	51	
4	76	65	53	
5	74	67	54	
6	75	66	53	
7	82	72	59	
8	83	64	52	
9	77	74	53	
10	86	73	54	
11	74	65	52	
12	86	68	57	
13	75	71	53	
14	85	71	55	
15	87	63	50	
16	80	80	57	
17	78	66	57	
18	86	72	57	
19	90	73	56	
20	95	56	45	
21	96	53	42	
22	108	46	38	
23	104	59	45	
24	112	63	49	
25	79	91	55	
26	81	94	55	
27	80	105	54	
28	87	96	55	
29	89	95	54	
30	74	56	56	
31	73	55	58	
32	68	48	58	
33	80	59	63	
34	85	62	64	
35	78	66	52	

36	79	68	54
37	59	40	57
38	58	39	57
39	52	31	56
40	67	44	63
41	76	51	71
42	70	82	60
43	77	87	65
44	81	88	66
45	76	115	53
46	78	116	54
47	74	124	51
48	83	118	56
49	87	121	58
50	96	76	42
51	103	79	42
52	111	86	50
53	95	52	55
54	103	56	59
55	112	60	64
56	121	45	40
57	121	48	39

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJDB.RGB;3

58	134	41	38
59	129	48	40
60	140	53	45
61	151	30	35
62	150	31	35
63	163	31	38
64	157	43	43
65	165	54	52
66	120	66	37
67	127	74	42
68	138	82	49
69	84	71	56
70	84	72	57
71	119	40	48
72	127	46	53
73	138	54	60
74	92	44	63
75	100	50	68
76	108	54	76
77	93	73	55
78	109	83	63
79	95	98	42
80	103	103	46
81	110	109	48
82	69	132	46
83	70	132	49
84	66	137	45
85	76	133	53
86	80	137	55
87	69	104	63
88	74	106	67
89	81	114	74
90	61	64	62
91	69	71	68
92	78	79	78
93	55	32	71
94	57	33	73
95	57	34	80
96	71	46	87
97	80	57	94

98	61	38	96
99	62	40	96
100	63	44	108
101	70	47	104
102	76	49	107
103	87	71	55
104	87	72	55
105	58	63	80
106	67	72	89
107	74	78	97
108	56	87	63
109	69	100	76
110	77	111	90
111	67	129	65
112	71	131	69
113	79	139	77
114	65	145	47
115	63	146	43
116	60	156	41
117	72	152	50
118	80	163	60
119	93	120	41

_SRV2$DUA1:[USERS.SMART.KEIDAR_M]DB.RGB;3

120	101	125	46
121	110	133	53
122	94	97	59
123	111	110	69
124	91	66	66
125	108	78	79
126	89	37	77
127	98	44	85
128	106	47	91
129	118	42	66
130	126	46	70
131	138	55	82
132	119	68	54
133	138	82	65
134	120	93	43
135	127	97	43
136	138	108	51
137	89	70	53
138	89	70	53
139	150	67	43
140	158	77	49
141	168	86	55
142	149	34	50
143	158	47	59
144	166	58	68
145	176	39	43
146	179	35	40
147	191	35	41
148	186	51	50
149	192	59	56
150	199	35	41
151	202	37	46
152	211	38	48
153	205	51	50
154	211	59	52
155	177	68	45
156	188	77	54
157	194	88	60
158	174	41	58
159	182	51	63

160	190	61	70
161	147	41	68
162	155	49	74
163	163	57	83
164	149	69	59
165	166	88	75
166	151	93	49
167	159	99	51
168	165	112	56
169	118	118	40
170	125	124	45
171	89	70	53
172	89	71	53
173	138	137	56
174	118	93	56
175	139	110	70
176	115	63	64
177	138	82	83
178	114	37	82
179	124	46	89
180	136	52	95
181	88	38	98

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJDB.RGB;3

182	99	47	105
183	106	50	112
184	88	60	82
185	108	78	101
186	92	93	76
187	110	108	89
188	93	126	66
189	109	135	75
190	89	141	41
191	101	152	50
192	109	159	58
193	61	165	47
194	64	170	50
195	61	181	50
196	70	176	57
197	78	181	60
198	63	150	66
199	71	160	77
200	77	167	83
201	62	124	77
202	68	132	85
203	74	138	94
204	60	91	85
205	91	72	54
206	91	71	53
207	67	98	98
208	73	106	105
209	58	67	101
210	67	72	109
211	74	80	119
212	62	47	116
213	60	45	120
214	58	41	126
215	70	51	125
216	77	54	131
217	60	43	135
218	56	38	137
219	58	42	148
220	69	51	143
221	76	56	150

222	58	66	123
223	68	76	132
224	77	84	137
225	62	93	112
226	72	101	122
227	78	110	131
228	63	124	105
229	69	129	111
230	73	135	118
231	60	150	85
232	67	161	97
233	75	168	110
234	63	175	69
235	69	180	77
236	72	184	82
237	62	187	49
238	59	190	45
239	94	72	55
240	94	72	52
241	57	204	49
242	68	195	57
243	75	201	64

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJDB.RGB;3

244	89	165	44
245	99	171	53
246	109	178	63
247	92	153	71
248	108	162	82
249	89	123	81
250	108	135	100
251	89	88	92
252	108	105	109
253	89	60	104
254	107	77	122
255	90	38	120
256	98	42	127
257	106	47	132
258	115	35	101
259	124	45	109
260	134	52	114
261	117	62	88
262	137	85	105
263	118	88	69
264	140	107	89
265	119	120	60
266	141	134	77
267	120	145	48
268	128	152	55
269	136	159	60
270	151	122	49
271	158	130	55
272	166	137	62
273	98	71	53
274	99	72	53
275	153	94	61
276	168	116	75
277	151	68	70
278	165	87	87
279	147	35	85
280	155	46	72
281	164	57	99
282	176	39	70
283	184	48	77

284	191	57	86
285	177	70	73
286	194	91	76
287	187	92	52
288	190	103	60
289	195	121	62
290	200	70	46
291	205	83	50
292	211	93	57
293	199	37	58
294	205	48	64
295	210	59	71
296	219	39	46
297	224	31	41
298	235	28	41
299	226	49	50
300	233	56	56
301	239	34	48
302	247	31	46
303	249	27	45
304	254	30	45
305	254	36	46

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJDB.RGB;3

306	254	29	48
307	103	76	58
308	104	75	57
309	254	28	43
310	250	45	50
311	254	40	40
312	250	20	46
313	254	23	44
314	251	40	49
315	254	42	47
316	254	49	43
317	254	30	49
318	251	45	53
319	254	60	60
320	220	76	49
321	227	84	55
322	233	96	61
323	220	39	61
324	227	50	70
325	234	56	75
326	199	37	72
327	205	50	78
328	211	59	88
329	198	71	64
330	210	94	76
331	199	98	47
332	203	111	54
333	210	126	61
334	177	124	50
335	186	132	55
336	191	142	60
337	177	95	65
338	190	121	82
339	173	74	78
340	187	94	92
341	105	79	61
342	106	80	61
343	172	36	90
344	181	46	96
345	188	57	102

346	142	33	103
347	151	44	109
348	161	53	116
349	148	66	89
350	163	86	105
351	151	95	76
352	165	116	96
353	152	125	64
354	167	141	81
355	151	145	51
356	157	153	56
357	165	161	62
358	118	164	47
359	126	172	54
360	137	178	61
361	119	147	63
362	137	159	81
363	117	120	78
364	138	137	100
365	115	86	91
366	135	110	111
367	114	59	105

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJDB.RGB;3

368	133	83	124
369	114	33	120
370	122	43	126
371	133	57	133
372	89	44	138
373	97	52	144
374	103	55	151
375	108	83	63
376	108	82	64
377	85	61	124
378	104	79	139
379	84	85	111
380	105	107	131
381	85	121	104
382	104	132	120
383	86	152	88
384	103	164	107
385	88	174	71
386	105	182	85
387	88	187	48
388	98	193	56
389	107	201	62
390	59	210	54
391	57	214	51
392	56	216	49
393	65	213	50
394	56	221	49
395	58	222	48
396	55	221	51
397	57	221	52
398	65	218	50
399	55	217	58
400	64	215	58
401	57	222	60
402	65	222	61
403	69	221	52
404	57	224	66
405	69	215	64
406	72	218	67
407	61	184	75

408	65	199	83
409	112	86	68
410	111	86	67
411	69	208	90
412	56	180	95
413	63	184	101
414	69	187	113
415	56	150	110
416	63	158	121
417	71	164	126
418	56	124	124
419	66	130	132
420	74	136	139
421	58	95	132
422	67	101	140
423	75	110	149
424	57	69	142
425	67	78	150
426	75	85	157
427	58	41	153
428	56	39	154
429	54	39	155

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJDB.RGB;3

430	64	38	154
431	54	46	156
432	62	45	157
433	54	39	160
434	54	38	162
435	54	40	164
436	57	41	166
437	63	40	164
438	55	49	166
439	64	48	164
440	66	41	162
441	58	54	164
442	66	49	156
443	117	90	72
444	116	90	70
445	75	55	163
446	57	71	158
447	57	70	166
448	55	84	171
449	67	80	163
450	75	87	171
451	57	96	149
452	67	103	157
453	75	112	167
454	57	125	141
455	66	131	149
456	74	139	158
457	57	148	131
458	65	157	140
459	73	164	148
460	55	175	119
461	64	181	127
462	71	187	133
463	55	197	99
464	61	201	106
465	68	208	117
466	56	212	79
467	55	224	76
468	51	229	90
469	63	219	86

470	70	226	93
471	87	206	52
472	83	218	52
473	99	216	53
474	97	211	59
475	104	218	64
476	85	190	72
477	122	97	76
478	122	94	73
479	105	205	67
480	85	176	94
481	100	184	111
482	85	150	114
483	102	160	129
484	87	117	123
485	107	131	143
486	88	87	132
487	107	107	152
488	89	64	143
489	107	80	160
490	88	37	156
491	85	34	165

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJDB.RGB;3

492	100	32	165
493	98	45	162
494	107	51	163
495	115	32	137
496	122	42	143
497	134	52	151
498	114	60	125
499	133	84	141
500	115	86	113
501	135	111	132
502	116	117	101
503	136	136	118
504	120	148	86
505	135	163	102
506	119	166	66
507	136	181	85
508	117	185	51
509	123	192	58
510	133	199	64
511	126	99	78
512	127	98	78
513	147	167	52
514	155	175	60
515	163	183	67
516	150	149	67
517	164	164	85
518	152	123	83
519	163	143	102
520	150	93	97
521	162	117	116
522	148	66	108
523	163	87	126
524	148	34	119
525	157	45	124
526	164	55	134
527	172	36	107
528	181	47	115
529	189	58	121
530	173	71	96
531	187	92	110

532	176	99	82
533	184	124	101
534	179	130	70
535	192	147	83
536	174	149	51
537	184	159	57
538	192	168	64
539	199	134	56
540	205	141	58
541	211	149	64
542	200	105	69
543	210	129	83
544	197	74	80
545	133	101	82
546	134	101	81
547	212	96	98
548	197	36	91
549	204	49	99
550	211	59	106
551	220	37	75
552	227	48	83
553	235	59	93

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJDB.RGB;3

554	220	75	65
555	233	99	81
556	221	108	51
557	226	116	56
558	233	127	59
559	242	72	51
560	254	70	47
561	254	91	49
562	249	87	59
563	254	101	63
564	242	33	66
565	254	32	61
566	254	33	71
567	250	48	75
568	254	60	77
569	242	35	80
570	254	34	79
571	254	37	88
572	250	51	88
573	254	64	91
574	240	76	67
575	247	91	75
576	254	106	79
577	237	108	52
578	253	108	54
579	139	108	91
580	140	108	90
581	254	124	53
582	247	119	59
583	254	133	63
584	218	137	54
585	225	145	58
586	231	156	62
587	219	108	70
588	232	132	84
589	220	75	82
590	233	99	102
591	219	39	97
592	227	50	103
593	233	59	110

594	197	36	110
595	204	51	120
596	210	61	127
597	197	74	100
598	210	95	116
599	199	103	85
600	211	129	104
601	199	134	71
602	211	152	85
603	198	156	54
604	204	165	60
605	211	176	65
606	174	171	53
607	182	180	60
608	190	189	65
609	175	153	73
610	189	170	87
611	173	133	88
612	187	147	107
613	143	117	99
614	143	115	98
615	172	101	103

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJDB.RGB;3

616	186	122	119
617	170	69	111
618	188	93	129
619	170	34	121
620	180	46	130
621	188	56	137
622	147	34	134
623	154	45	143
624	162	54	152
625	145	67	127
626	161	86	144
627	146	94	119
628	159	116	137
629	147	126	109
630	159	142	123
631	148	149	92
632	158	168	109
633	148	170	75
634	162	185	93
635	145	185	55
636	154	194	63
637	161	204	68
638	115	205	56
639	114	218	54
640	127	220	54
641	123	213	63
642	132	221	68
643	116	189	74
644	132	203	94
645	114	171	95
646	126	191	113
647	145	121	103
648	144	122	103
649	114	145	115
650	127	166	129
651	114	115	127
652	133	133	145
653	113	89	135
654	134	110	156
655	113	65	140

656	134	84	163
657	113	33	154
658	113	29	164
659	128	29	164
660	123	43	161
661	135	51	169
662	89	66	161
663	100	76	167
664	107	81	176
665	91	92	152
666	107	108	171
667	88	121	145
668	107	132	163
669	86	144	133
670	102	157	155
671	84	172	122
672	99	184	139
673	83	193	102
674	100	208	121
675	86	212	79
676	95	218	87
677	103	225	93

_SRV2$DUA1:[USERS.SMART.KEIDAR_M1DB.RGB;3

678	55	221	105
679	51	232	106
680	51	232	122
681	148	130	109
682	147	132	110
683	61	224	116
684	67	233	126
685	53	199	128
686	61	205	134
687	67	213	144
688	56	176	141
689	63	183	152
690	70	190	161
691	58	154	152
692	66	162	161
693	70	168	171
694	60	130	160
695	68	136	170
696	74	140	181
697	62	99	167
698	60	99	179
699	60	113	185
700	71	103	175
701	77	113	188
702	59	130	180
703	56	131	192
704	54	143	201
705	66	136	190
706	73	141	199
707	59	157	172
708	66	165	184
709	71	171	192
710	57	178	161
711	62	186	170
712	64	194	177
713	55	200	148
714	61	208	159
715	151	136	112
716	150	136	113

717	65	218	158
718	51	222	134
719	48	236	138
720	46	237	148
721	59	228	142
722	66	233	148
723	83	216	102
724	93	222	110
725	100	228	122
726	82	196	128
727	97	209	145
728	85	173	143
729	99	183	161
730	88	148	155
731	102	164	170
732	90	122	164
733	108	134	180
734	91	93	169
735	101	99	178
736	108	106	188
737	115	63	161
738	123	74	170
739	135	80	180

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJDB.RGB;3

740	115	89	156
741	135	111	174
742	117	117	149
743	136	133	163
744	114	142	138
745	131	158	154
746	111	170	122
747	125	187	132
748	112	191	101
749	153	140	116
750	154	140	116
751	132	210	118
752	114	210	76
753	121	218	86
754	131	225	94
755	144	209	54
756	144	220	50
757	157	221	52
758	153	215	60
759	161	223	65
760	145	193	77
761	161	209	93
762	146	171	94
763	159	189	115
764	144	155	113
765	158	169	128
766	147	124	128
767	161	142	145
768	144	97	138
769	161	114	156
770	145	68	145
771	160	84	164
772	147	34	152
773	147	29	161
774	159	31	163
775	154	45	160
776	163	55	169
777	172	37	160
778	180	49	168
779	188	57	154

780	172	71	129
781	187	92	143
782	169	101	120
783	157	144	122
784	157	146	123
785	188	123	138
786	171	133	112
787	188	147	124
788	173	157	94
789	191	172	110
790	175	175	75
791	194	193	89
792	172	191	54
793	183	201	59
794	193	209	66
795	198	179	52
796	205	189	59
797	212	198	66
798	199	160	73
799	212	181	87
800	198	137	90
801	211	156	109

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJDB.RGB;3

802	198	106	106
803	211	130	122
804	197	74	118
805	209	96	135
806	196	38	126
807	203	50	134
808	209	59	142
809	219	39	114
810	225	51	121
811	232	59	129
812	219	78	105
813	232	99	119
814	219	111	90
815	230	133	109
816	216	142	78
817	161	147	128
818	161	147	128
819	232	160	89
820	219	164	57
821	226	175	60
822	234	187	63
823	239	144	49
824	254	146	49
825	254	160	50
826	248	156	57
827	254	165	64
828	240	114	70
829	248	126	78
830	254	139	82
831	242	79	85
832	250	96	93
833	254	111	99
834	243	37	98
835	254	36	95
836	254	36	102
837	251	55	107
838	254	65	105
839	241	37	111
840	254	35	110
841	254	34	119

842	248	50	121
843	254	65	126
844	240	77	105
845	248	94	111
846	254	107	116
847	239	114	90
848	247	127	100
849	254	138	107
850	235	145	74
851	166	150	134
852	167	152	134
853	247	159	86
854	254	168	94
855	238	172	51
856	254	178	54
857	254	190	55
858	246	183	61
859	254	194	69
860	220	189	56
861	228	199	61
862	234	204	66
863	220	168	75

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJDB.RGB;3

864	234	188	90
865	219	142	93
866	232	161	113
867	220	110	112
868	232	135	128
869	219	77	123
870	233	98	142
871	219	41	130
872	225	50	138
873	231	55	146
874	195	36	142
875	202	46	150
876	208	56	159
877	196	72	134
878	209	96	150
879	197	105	125
880	209	130	141
881	196	136	114
882	208	156	125
883	197	162	96
884	208	181	113
885	172	157	142
886	172	156	140
887	199	181	80
888	210	200	93
889	196	199	52
890	204	208	61
891	210	216	68
892	173	212	55
893	174	223	53
894	189	225	50
895	183	219	61
896	191	226	68
897	174	198	78
898	191	212	75
899	174	178	97
900	189	191	115
901	170	156	117
902	189	171	131
903	169	128	132

904	186	144	146
905	171	99	139
906	187	121	156
907	169	69	146
908	186	86	163
909	169	34	152
910	171	31	161
911	182	32	166
912	177	47	162
913	185	56	170
914	144	67	165
915	152	76	170
916	158	85	178
917	143	99	158
918	159	115	172
919	176	161	144
920	176	162	144
921	146	123	151
922	159	143	163
923	139	152	136
924	156	171	152
925	139	180	119

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJDB.RGB;3

926	155	194	136
927	143	195	99
928	156	213	121
929	144	213	79
930	151	220	87
931	158	227	97
932	114	214	100
933	121	222	111
934	130	229	122
935	107	195	131
936	129	213	142
937	111	167	148
938	127	188	159
939	115	140	159
940	133	164	168
941	115	118	167
942	133	138	179
943	115	90	172
944	123	97	180
945	133	111	188
946	89	123	185
947	98	130	191
948	105	138	198
949	87	152	175
950	101	168	187
951	82	176	163
952	96	191	178
953	180	166	149
954	180	167	150
955	80	200	154
956	94	216	168
957	79	220	134
958	89	226	141
959	96	233	150
960	51	226	159
961	47	242	164
962	46	246	176
963	58	232	166
964	64	241	173
965	52	206	175

966	57	214	182
967	62	224	191
968	52	184	186
969	60	190	197
970	66	197	207
971	58	158	195
972	54	159	211
973	52	172	221
974	64	164	206
975	70	170	218
976	52	183	210
977	48	186	232
978	46	201	242
979	61	193	216
980	65	197	237
981	50	213	198
982	56	222	212
983	60	229	224
984	48	234	181
985	45	250	186
986	42	254	198
987	184	172	153

_SRV2$DUA1:[USERS.SMART.KEIDAR_M3DB.RGB;3

988	184	173	154
989	56	241	190
990	62	251	198
991	76	222	158
992	86	230	165
993	94	236	170
994	78	201	173
995	93	219	187
996	81	177	185
997	95	191	203
998	86	151	195
999	94	158	204
1000	100	166	215
1001	117	119	182
1002	124	130	189
1003	133	140	198
1004	117	148	173
1005	131	169	189
1006	112	168	165
1007	126	192	176
1008	107	194	155
1009	123	216	165
1010	109	218	134
1011	117	226	140
1012	128	233	145
1013	142	216	104
1014	147	223	114
1015	155	230	124
1016	140	199	127
1017	155	215	143
1018	140	179	145
1019	153	195	157
1020	142	148	157
1021	188	174	156
1022	188	175	157
1023	157	173	172
1024	146	123	168
1025	158	144	181
1026	146	92	174
1027	158	88	181

1028	160	114	189
1029	170	67	163
1030	179	76	171
1031	186	84	179
1032	170	95	158
1033	188	117	174
1034	170	128	151
1035	187	145	164
1036	169	154	140
1037	188	173	153
1038	172	175	122
1039	182	191	139
1040	174	198	102
1041	188	210	121
1042	174	213	80
1043	182	219	92
1044	189	226	100
1045	197	216	57
1046	199	227	53
1047	210	231	52
1048	202	224	62
1049	209	231	69

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJDB.RGB;3

1050	198	202	78
1051	209	215	95
1052	199	185	100
1053	208	197	115
1054	196	160	117
1055	190	178	160
1056	191	177	159
1057	210	182	131
1058	196	136	131
1059	210	157	144
1060	196	106	140
1061	209	131	156
1062	196	71	149
1063	210	97	166
1064	195	34	157
1065	197	29	166
1066	210	29	166
1067	201	46	164
1068	209	54	172
1069	218	36	144
1070	225	45	153
1071	233	51	163
1072	219	76	138
1073	234	97	156
1074	219	109	130
1075	233	131	146
1076	220	142	118
1077	232	159	133
1078	219	167	98
1079	234	183	119
1080	220	190	78
1081	234	203	97
1082	220	205	56
1083	225	215	61
1084	233	218	70
1085	239	194	55
1086	254	198	61
1087	254	205	58
1088	244	199	62
1089	195	183	166

1090	195	182	164
1091	254	206	71
1092	239	174	73
1093	246	183	84
1094	254	194	95
1095	240	148	95
1096	243	160	104
1097	254	169	115
1098	241	116	112
1099	249	127	120
1100	254	139	126
1101	240	77	124
1102	249	94	132
1103	254	105	137
1104	240	37	127
1105	254	35	125
1106	254	34	135
1107	250	50	136
1108	254	61	140
1109	241	33	146
1110	254	31	144
1111	254	28	152

_SRV2$DUA1:[USERS.SMART.KEIDAR_HODB.RGB;3

1112	249	45	153
1113	254	56	162
1114	239	74	140
1115	248	91	150
1116	254	102	158
1117	240	113	133
1118	247	124	141
1119	254	137	149
1120	238	148	122
1121	245	158	129
1122	254	165	135
1123	197	185	170
1124	198	185	169
1125	239	175	100
1126	248	186	110
1127	254	193	121
1128	239	195	77
1129	247	201	69
1130	254	209	100
1131	239	211	53
1132	254	218	54
1133	254	221	51
1134	247	217	62
1135	254	222	72
1136	217	222	53
1137	221	233	52
1138	232	234	50
1139	224	228	62
1140	232	232	69
1141	220	206	80
1142	232	220	101
1143	218	191	99
1144	233	200	124
1145	218	168	121
1146	232	181	140
1147	218	145	137
1148	231	157	153
1149	238	111	146
1150	231	130	163
1151	217	75	155

1152	231	95	173
1153	216	69	161
1154	221	31	169
1155	236	26	190
1156	220	55	187
1157	201	191	176
1158	201	190	176
1159	231	49	176
1160	194	70	165
1161	201	80	174
1162	208	94	181
1163	195	103	157
1164	208	129	174
1165	194	137	149
1166	208	156	161
1167	195	161	136
1168	207	181	153
1169	194	183	123
1170	207	197	138
1171	195	201	100
1172	208	215	122
1173	196	218	86

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJOB.RGB]3

1174	201	225	88
1175	207	230	96
1176	169	218	108
1177	179	224	116
1178	185	230	125
1179	165	199	131
1180	186	213	146
1181	168	179	143
1182	186	194	161
1183	169	154	156
1184	186	171	174
1185	171	124	166
1186	185	144	183
1187	169	93	174
1188	178	104	181
1189	184	118	190
1190	143	124	179
1191	204	194	181
1192	204	194	182
1193	150	134	189
1194	154	143	199
1195	142	152	172
1196	153	171	191
1197	137	179	164
1198	153	196	178
1199	137	202	151
1200	154	219	166
1201	138	221	132
1202	146	227	141
1203	154	232	149
1204	103	220	156
1205	111	228	163
1206	121	237	170
1207	104	199	171
1208	121	220	187
1209	108	177	182
1210	124	195	203
1211	112	151	192
1212	116	160	202
1213	124	168	213

1214	81	179	210
1215	88	186	221
1216	92	193	231
1217	75	205	195
1218	89	223	217
1219	75	227	178
1220	83	235	185
1221	89	241	195
1222	45	241	210
1223	37	254	221
1224	36	254	236
1225	207	197	186
1226	207	197	185
1227	49	249	222
1228	53	254	235
1229	45	213	229
1230	40	221	254
1231	37	239	254
1232	52	222	243
1233	55	231	254
1234	39	246	247
1235	32	254	254

_SRV2$DUA1:[USERS.SHART.KEIDAR_MJDB.RGB;3

1236	34	254	254
1237	33	254	254
1238	40	254	254
1239	31	254	254
1240	39	254	254
1241	32	254	254
1242	40	254	254
1243	32	254	254
1244	32	254	254
1245	33	254	254
1246	34	254	254
1247	40	254	254
1248	45	252	254
1249	43	254	254
1250	50	254	254
1251	70	235	207
1252	80	241	216
1253	84	247	227
1254	75	208	222
1255	82	215	235
1256	86	220	243
1257	106	176	200
1258	110	184	207
1259	211	202	191
1260	211	202	191
1261	121	194	228
1262	102	203	191
1263	118	219	213
1264	101	224	175
1265	107	232	183
1266	119	239	191
1267	138	223	155
1268	146	230	163
1269	154	235	170
1270	135	203	170
1271	152	218	187
1272	139	177	179
1273	153	194	200
1274	142	152	191
1275	147	161	208

1276	154	171	214
1277	170	128	185
1278	178	137	194
1279	187	147	203
1280	169	157	177
1281	186	175	195
1282	168	181	165
1283	186	195	182
1284	168	204	152
1285	186	216	168
1286	168	221	135
1287	178	226	141
1288	185	233	147
1289	196	220	107
1290	201	225	116
1291	206	230	126
1292	191	200	127
1293	214	206	196
1294	214	207	196
1295	209	215	147
1296	195	184	145
1297	208	198	162

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJDB.RGB;3

1298	194	162	156
1299	209	177	173
1300	194	138	168
1301	212	157	184
1302	197	104	173
1303	203	116	181
1304	212	128	191
1305	218	74	171
1306	226	85	179
1307	233	98	184
1308	219	112	161
1309	233	130	179
1310	218	143	152
1311	233	156	169
1312	219	170	141
1313	232	179	161
1314	219	188	126
1315	233	199	146
1316	219	207	104
1317	232	220	129
1318	218	222	82
1319	225	228	94
1320	232	234	105
1321	239	224	55
1322	254	226	52
1323	244	234	50
1324	247	228	48
1325	254	227	47
1326	242	232	49
1327	216	211	202
1328	217	211	202
1329	254	236	46
1330	254	236	43
1331	254	235	48
1332	254	236	49
1333	249	228	58
1334	254	229	61
1335	249	235	60
1336	254	238	59
1337	249	228	67

1338	254	239	65
1339	254	237	72
1340	241	210	83
1341	248	216	96
1342	254	224	110
1343	242	196	107
1344	249	201	118
1345	254	205	130
1346	242	173	129
1347	251	183	136
1348	254	190	142
1349	241	148	141
1350	250	158	151
1351	254	166	158
1352	241	113	153
1353	250	126	160
1354	254	139	168
1355	242	76	161
1356	250	90	170
1357	254	100	177
1358	241	32	163
1359	254	29	163

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJDB.RGB]3

1360	244	28	170
1361	218	213	206
1362	218	213	206
1363	252	25	163
1364	254	25	163
1365	252	41	170
1366	254	42	169
1367	253	22	167
1368	254	24	167
1369	254	21	164
1370	254	27	169
1371	254	27	169
1372	253	39	175
1373	254	39	175
1374	252	44	173
1375	254	47	179
1376	254	60	182
1377	242	72	178
1378	254	67	189
1379	250	90	187
1380	254	92	193
1381	254	98	194
1382	242	110	169
1383	250	122	178
1384	254	135	186
1385	240	146	159
1386	248	156	167
1387	254	163	173
1388	240	171	149
1389	248	178	157
1390	254	187	163
1391	240	192	133
1392	248	197	139
1393	254	203	146
1394	236	207	107
1395	220	217	211
1396	221	217	211
1397	247	218	123
1398	254	224	135
1399	238	224	87

1400	254	241	85
1401	242	231	102
1402	254	243	108
1403	254	243	113
1404	218	223	108
1405	224	229	119
1406	231	234	131
1407	217	205	134
1408	233	220	151
1409	217	190	150
1410	232	199	167
1411	218	171	160
1412	231	179	181
1413	218	143	170
1414	232	155	189
1415	219	111	177
1416	225	119	185
1417	232	130	194
1418	197	136	189
1419	205	147	198
1420	212	158	206
1421	195	162	180

_SRV2$DUA1:[USERS.SMART.KEIDAR_HJDB.RGB;3

1422	210	180	197
1423	195	182	166
1424	208	197	184
1425	194	203	153
1426	206	216	167
1427	194	220	140
1428	199	224	143
1429	228	223	217
1430	228	223	216
1431	208	233	153
1432	170	224	159
1433	178	229	165
1434	184	235	170
1435	168	206	173
1436	184	219	188
1437	168	183	185
1438	185	197	203
1439	168	158	195
1440	179	168	208
1441	185	177	214
1442	139	179	204
1443	147	187	214
1444	153	194	228
1445	138	204	191
1446	152	220	214
1447	140	226	177
1448	146	232	187
1449	153	236	193
1450	105	228	201
1451	111	233	212
1452	119	242	224
1453	105	204	220
1454	109	213	235
1455	118	219	247
1456	74	235	241
1457	64	254	254
1458	80	240	251
1459	78	250	254
1460	84	247	254

1462	87	[illegible]	254
1463	233	227	220
1464	232	227	220
1465	103	239	243
1466	103	252	254
1467	109	251	254
1468	132	228	200
1469	139	234	210
1470	146	240	221
1471	134	204	217
1472	142	211	230
1473	147	220	242
1474	167	184	204
1475	177	191	213
1476	185	197	224
1477	166	205	192
1478	183	220	212
1479	168	226	179
1480	175	230	186
1481	183	236	193
1482	195	223	159
1483	203	229	167

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJDB.RGB;3

1484	210	234	174
1485	196	204	175
1486	208	219	193
1487	196	185	190
1488	208	200	208
1489	196	164	199
1490	204	174	207
1491	211	181	216
1492	218	142	189
1493	225	151	197
1494	231	155	201
1495	212	163	175
1496	229	177	174
1497	237	229	224
1498	238	229	224
1499	215	186	170
1500	229	200	185
1501	216	204	158
1502	233	223	171
1503	216	223	140
1504	223	230	146
1505	232	238	152
1506	240	227	119
1507	254	245	123
1508	247	234	128
1509	254	246	134
1510	254	246	137
1511	240	210	140
1512	247	218	148
1513	254	225	155
1514	240	192	155
1515	246	200	164
1516	254	208	172
1517	240	170	167
1518	247	179	176
1519	254	188	185
1520	240	147	176
1521	248	156	184
1522	234	163	192
1523	240	111	193

1524	234	109	197
1525	248	122	181
1526	254	129	200
1527	254	135	199
1528	240	144	191
1529	254	146	201
1530	243	154	198
1531	241	232	227
1532	242	233	228
1533	254	159	207
1534	254	160	205
1535	237	168	183
1536	246	179	192
1537	253	191	194
1538	237	189	194
1539	245	200	182
1540	253	209	180
1541	238	211	161
1542	246	219	168
1543	254	228	176
1544	239	229	144
1545	254	248	148

_SRV2$DUA1:[USERS.SMART.KEIDAR_HJDB.RGB;3

1546	247	236	150
1547	254	248	157
1548	254	247	159
1549	218	226	164
1550	226	235	169
1551	232	241	175
1552	217	207	178
1553	228	221	192
1554	218	188	191
1555	230	204	206
1556	219	169	200
1557	225	175	207
1558	232	181	216
1559	196	187	208
1560	202	193	216
1561	207	198	226
1562	195	208	196
1563	208	222	215
1564	191	223	177
1565	245	237	234
1566	246	237	235
1567	200	232	189
1568	208	238	199
1569	164	226	201
1570	173	234	209
1571	182	239	218
1572	165	205	217
1573	175	211	226
1574	182	219	240
1575	130	227	231
1576	123	248	254
1577	135	237	243
1578	139	246	254
1579	144	244	254
1580	161	227	228
1581	161	243	254
1582	170	234	239
1583	177	243	251
1584	177	243	252
1585	193	297	206

1586	204	235	213
1587	210	239	223
1588	193	207	219
1589	200	212	229
1590	207	218	238
1591	217	190	210
1592	223	196	219
1593	231	204	227
1594	214	209	199
1595	227	223	216
1596	215	229	183
1597	221	234	189
1598	226	239	196
1599	246	243	242
1600	245	243	241
1601	239	232	167
1602	254	249	169
1603	245	239	173
1604	254	250	179
1605	254	250	180
1606	236	212	180
1607	244	220	188

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJDR.RGB;3

1608	254	229	197
1609	237	193	192
1610	246	204	196
1611	252	212	201
1612	239	170	202
1613	254	175	209
1614	247	179	208
1615	254	190	212
1616	253	188	214
1617	238	193	211
1618	254	200	224
1619	248	203	217
1620	254	210	233
1621	254	210	234
1622	235	212	199
1623	247	221	207
1624	254	230	214
1625	237	231	184
1626	254	251	192
1627	248	241	193
1628	254	253	201
1629	254	253	202
1630	214	228	207
1631	221	236	214
1632	230	246	220
1633	247	246	247
1634	247	246	247
1635	215	209	220
1636	223	216	228
1637	227	219	237
1638	190	227	226
1639	188	244	251
1640	197	234	237
1641	202	243	249
1642	201	243	251
1643	214	227	223
1644	213	242	250
1645	219	235	237
1646	223	245	249
1647	224	244	245

1648	234	229	205
1649	254	252	212
1650	246	240	212
1651	254	252	223
1652	254	251	225
1653	236	212	218
1654	254	212	236
1655	244	215	226
1656	254	217	240
1657	254	219	242
1658	235	230	226
1659	242	240	230
1660	254	238	251
1661	245	254	253
1662	245	239	237
1663	246	238	236
1664	250	237	233
1665	242	250	237
1666	242	241	230
1667	246	247	247
1668	246	247	247
1669	246	240	249

_SRV2$DUA1:[USERS.SMART.KEIDAR_MJDB.RGB;3

1670	252	240	249
1671	245	252	252
1672	246	246	246
1673	246	247	247
1674	246	247	248
1675	246	247	247
1676	246	246	248

==

APPENDIX E

TRANS/4 USER GUIDE

Important Notice

Restrictions on the use of this document:

This document is delivered subject to the following conditions and restrictions:

This documentation contains proprietary information of SCITEX CORPORATION, LTD. Such information is hereby supplied solely for the purpose of assisting explicitly and properly authorized users of the RESPONSE series of SCITEX systems in the proper operation of SCITEX equipment. Without the express prior written permission of SCITEX, no part of the contents hereof may be used for any other purpose, disclosed to any person or firm, or reproduced by any means.

The text and drawings herein are for the purposes of illustration and reference only. The specifications on which they are based are subject to change without notice.

Document No.771-9A005

Preliminary: December 1990

Trademarks:

Color Key®, Cromalin®, Agfaproof™

1. TRANS/4 OPERATION

1.1 INTRODUCTION TO TRANS/4

TRANS/4 is used for color proofing on the Iris Printer and consists of software that is resident on a Scitex workstation.

The Iris printer, together with the Scitex workstation, consistently gives an accurate reproduction of the press system which would be used on a final printing job. TRANS/4 sends picture data files from the workstation directly to the Iris printer, eliminating the need to generate films for each separation. Instead of spending several hours producing a color proof from films (e.g. Cromalin, Matchprint, etc.), the Iris printer produces a precise proof in only minutes and at a fraction of the cost.

FACTORY-SET CALIBRATIONS

TRANS/4 has been calibrated at the factory to simulate several proofing and printing systems. A specific list of these systems is provided with the magnetic tape which is supplied with the TRANS/4 system.

For every factory-calibrated system, a corresponding Color Transform file (type NDL) is supplied on the magnetic tape. When applying a particular Color Transform to a page, a color match of the target printing system is obtained. Proofing is a matter of simply specifying the appropriate Color Transform file when printing to the Iris.

CUSTOM CALIBRATIONS

One of the powerful features of TRANS/4 is its capability of being calibrated to most any printing environment. The user can optionally create his own Color Transforms in order to simulate a non-standard (not supplied by Scitex) press system. To do this, the Custom-Calibration Procedure (Chapter 3.) must be performed.

The Calibration Procedure

The calibration procedure is required once per target printing process and not each time a proof is output. As long as the final printing process or conditions do not change, the system is calibrated for that process (see Chapter 3. Custom Calibration).

Calibration consists of several procedures which require the use of a colorimeter. A reference pattern is printed on the target press system via films as well as directly to the Iris printer. The goal of calibration is to produce a proof which matches the press print. These two outputs are measured precisely and data is entered into the Scitex workstation. TRANS/4 is then able to create the appropriate Color Transform.

Once the calibration procedure is complete and there is a proper match, the system need not be calibrated for that printing process again. TRANS/4 is capable of storing virtually an unlimited number of sets of calibration data which reflect different printing processes. By specifying the desired data set, the Iris can simulate, on-the-fly, any printing process which has been previously calibrated.

1.2 PRINTING ON THE IRIS PRINTER

To output Scitex files onto the Iris printer, a command originating from the Iris Front-End-Processor (FEP) must first be sent to the Whisper workstation. This makes sure that the FEP is ready and waiting to receive the data to be printed.

Next, from the Whisper workstation, the command to begin proofing is sent to the FEP. A CT, LW or Final Layout file could be printed. A Color Transform file could also be specified so that the proof obtained, simulates some target printing method.

STEP 1. THE IRIS PRINTER

Place a sheet of paper onto the drum of the Iris Color Printer. (Read your Iris Operator's Manual for details.) Close the hood and put the printer on-line.

STEP 2. THE IRIS FEP

From the Iris FEP keyboard, select I/O AND FILES. Then select Pipe. This command ensures that the FEP is ready to receive data. Make sure that CLT and HUE fields have been disabled via the EDIT selection. The following screen appears:

Enter the following parameters ,if necessary, and continue to step 3.

Enable Scaling

Change this field to YES in order to scale the file on the fly. Usually, this field would be set to NO and any scaling will be done on the Scitex workstation.

Title, Subtitle

These fields allow you to add two lines of labels outside the print area of the proof. Enter information here such as the file name, color transformation, date, etc.

You may enter Edit before selecting I/O and Files in order to change parameters such as Offset, etc.

STEP 3. THE SCITEX STATION

On the Whisper, Select **Start Work** and choose the name of the file to be proofed. From the pop-up menu, select **Proof** to proof to the Iris printer. LW, CT and Final Layout files may be proofed Note, that the Color Transform file (NDL) must be located either in the "Public" table or in the current Job and Page you are working on.

Enter the parameters which are pertinent to your printing requirements and select the **Proof** softkey to initiate the printing. At this point you may continue your other work and the printing will continue in the background. When printing is complete a message will be flashed on your screen.

PARAMETERS TO BE ENTERED

NDL

Type the name of the Color Transform to apply to the file to be printed, if any.

Excurve File

If you do not use the factory default, type the name of the Excurve which has been created. (Refer to Chapter 3, Custom Calibration.)

PROOF

Pressing this softkey sends the page data to the Iris printer, via the FEP.

At the FEP, you will be prompted to press ENTER .

QUIT

Pressing this softkey aborts the operation.

Paper Size

The paper size is fixed to 600 x 600 mm for Iris 3024; 863 x 1189 mm for Iris 3047; and 305 x 457 mm for Iris Smartjet 4012.

File Size

These fields display the height and width of the selected file.

Output Size

The default output size is the size of the selected file. Change these values in order to scale the picture up and down. Output size may not exceed the paper size. Changing this field changes the Scale Factor field.

Scale Factor

To enlarge or reduce the picture. Changing this field changes the Output Size field.

Gradation

Type the name of the gradation curve to be applied to the file, if any.

MTX

Type the name of the Matrix to be applied to the file, if any.

2. TRANS/4 INSTALLATION

2. 1 HARDWARE

TRANS/4 requires the following equipment:

- Any Whisper Cabinet with four available slots to accommodate the following required boards:
  - HSP (High Speed Processor - 2 boards)
  - PM (Picture Memory) Board
  - GPIB Board (Handshake)
  - SCD Board (attaches to CPU board)
- Iris 3024/3047/4012 Color Ink Jet Proofer
- Iris Front-End-Processor (FEP)
- X-Rite 918F Colorimeter (optional)
- Cable for X-Rite - SCD connection

The figure on the next page shows how the equipment is connected. The Iris Color Ink Jet Proofer is connected to the Iris front-end-processor (FEP) as explained in the Iris Operator's Manual.

The Iris FEP is linked to the Whisper by way of the GPIB board and cable.

The X-Rite colorimeter connects to the Whisper via a cable to the SCD board.

NOTE:

- When using the X-RITE Colorimeter connected to the SCD, make sure FCO #12637 was performed.
- Refer to Appendix I, HSP Installation Procedure for installation instructions for the High Speed Processor Board.

Figure 1. TRANS/4 Hardware Configuration

2.2 SYSTEM CONFIGURATION

When performing System Configuration, the TRANS/4 option must be enabled along with assigning the printer logical name (Iris) to a physical GPIB channel number. All this is accomplished through the following series of dialogues:

First, press **System, Services and Configuration** softkeys from the I/O workstation in order to access the System Configuration Main Menu.

From the System Configuration Main Menu, press the **Secur. Info** softkey and the following screen appears:

The password for your system must be obtained from your Scitex Subsidiary Administrator. Be prepared to give him the serial # and workstation configuration. Set the options as they apply to your system, making sure that *HandShake* and *TRANS/4* are set to "YES". Press the **Main Menu** softkey to bring you back to the System Configuration Main Menu. From there press the **Station Config** and then **Handsk Config** softkeys to bring up the screen for setting up the Iris communications channel.

Press the GPIB softkey and the following screen appears:

Set the *GPIB Address 1* field to the channel number that the Iris printer is physically connected to on the GPIB board. Values may range from 1 - 4. Make sure the other parameters are set as shown.

The next step is to assign this physical channel number to the Iris logical name. To do this press the **Main Menu** softkey which brings you back to the System Configuration Main Menu. Press the **Logical Names** softkey and the following appears:

Press the **Handsk LogNames** key. This brings you to the screen on the next page:

Select the *device type* to be "GPIB" by toggling with the space bar. Select the physical channel number (as entered for GPIB configuration previously) under the *Channel Address*. The *Logical Name* should be set to "Iris". Now press the **Save & Exit** softkey to install all the parameters you have just set. The TRANS/4 system is now configured to run on your workstation.

2.3 COLOR BALANCE

The Color Balance procedure is done once for every Iris Installation. Its purpose is to initialize Iris Ink Jet color densities in order to calibrate your Iris printer to mimic the factory reference. **The procedure described in this section should be performed only if Factory-Set Color Transforms will be used when proofing.** If you intend to create your own Custom Color Transforms, skip this section and go on to Chapter 3. Custom Calibration.

The following is required for this procedure:

- The Factory-supplied proof of the sample pattern, referred to as the Color Wedge (included with this manual).
- The Color Wedge data file (COLWDG.P), resident on Magtape and supplied with TRANS/4.
- A densitometer (or colorimeter).

The outline of the Color Balance procedure is as follows:

- Measure the factory proof with your densitometer.
- Set the Iris Printer with density values of 100 for CMYB.
- Download the Color Wedge file from magtape to the Scitex station and create a proof of it on your Iris Ink Jet printer.
- Measure your proof and compare it to the factory proof measurements.
- Adjust the Iris density values accordingly.
- Create another proof of the Color Wedge on your Iris printer.
- Measure your proof. It should match the factory proof. If it doesn't, further adjustments may be necessary.

Color Balance - Detailed Description

The Color Wedge is the sample reference pattern which is used when setting the Color Balance on the Iris printer. With every TRANS/4 system, a factory proof of the Color Wedge is supplied. Note that this proof does not reflect the final printing quality and is used strictly for adjustment purposes. A magtape containing the file COLWDG.P is also supplied. Download this file into your workstation disk.

% Dot Values System values

1. Measure the Factory Proof

With your densitometer (or colorimeter), measure the factory proof at 100% Dot for each of the four separations. Measure Cyan density for

cyan; Magenta density for Magenta; Yellow density for Yellow and Black density for Black. Write these four values down.

2. Set the Iris densities to 100 for CMYB.

Set the color densities on the Iris Front-End-Processor (FEP) to 100 for C,M,Y and B. To enter the densities, on the FEP, select SET UP, **default configuration,** with CLT and HUE disabled. Then select EDIT, **Configuration** and then **Color Balance.**

Set the densities to 100 for Cyan, Magenta, Yellow and Black.

3. Create a proof of the Color Wedge on your Iris Printer

Make sure you have downloaded the Color Wedge file (COLWDG.P) from the magtape supplied with the system. Select the Proof option (as described in Chapter 1. TRANS/4 Operation), to print the Color Wedge on your system. Make sure you do not specify any gradation, MTX, NDL or Excurve file.

4. Measure your proof

With the densitometer, measure your proof starting with Cyan at 100% Dot. If your measurement here matches the factory proof measurement (as described in step 1.) then the Cyan density setting may remain at 100. If, however, it does not match, do the following: locate the position on your proof for Cyan where the densitometer reading is

the same as was measured from the factory proof at 100% Dot. This position on the scale represents the density which should be set. Write down the value and repeat the procedure for Magenta, Yellow and Black.

An Example

Suppose you measure the density of Cyan at 100% Dot on the factory proof and its value is 1.8. On your site's proof you locate density 1.8 for Cyan at the 80% Dot location on the Color Wedge. That means, you must set the Iris density for Cyan to be 80. The same procedure should be applied to the other colors.

5. Adjust Iris Densities

After all density values have been determined, enter them into the FEP as was described in Step 2.

6. Create another proof

Output the Color Wedge once again, using the new density settings.

7. Measure your new proof

Measure the 100%Dot value for C,M,Y and B on your proof. These values should now closely match the values measured on the factory proof. If they don't, try to further adjust the Iris density settings by repeating steps 5. and 6. until the measurements are approximately equal.

8. Measure intermediary values on your proof

Once you are satisfied with matching 100%Dot values of the factory proof and your site's proof, measure several intermediary points (between 0%Dot and 100%Dot) on both proofs. Verify that they are "close." If there are wide discrepancies, it is possible that the Excurve supplied with TRANS/4 is not adequate for your system. It may, therefore, be advisable to create a new Excurve. Refer to the *Creating an Optimal Excurve* section of Chapter 3. Custom Calibration, for instructions on how to create this Excurve.

3. CUSTOM CALIBRATION (Optional)

Calibrating TRANS/4 is essential only when simulating a printing system whose Color Transform has not been supplied by the factory. The complete list of factory-calibrated Color Transforms is provided with the magnetic tape which is supplied with the TRANS/4 system.

Overview

When calibrating, it is necessary to establish that the Iris proofing system will simulate the press system being used.

The calibration methods described in this chapter adjust color parameters so that the Iris proofing system will consistently give an accurate reproduction of the desired printing method. The three procedures necessary for custom calibrations are:

- Adjusting The Color Balance
- Creating An Optimal Excurve
- Creating A Custom Color Transform

3.1 ADJUSTING THE COLOR BALANCE

This is the Color Balance procedure which is done when creating a custom Color Transform. (It replaces the Color Balance Procedure described in Chapter 2. Installation.) Its purpose is to initialize specific color parameters in order to achieve optimal results when proofing. It is done only once. If different Color Transforms are needed to be created, this adjustment does not have to be re-done.

In order to adjust the color balance, a sample pattern called the Color Wedge is printed on the Iris. This is provided by Scitex on magtap. With the aid of a colorimeter, the luminances will be measured. This section gives you a step-by-step approach to entering the proper data into the system.

THE CIELAB COLOR SYSTEM

Since color is a matter of personal perception and therefore subjective, a method of objectively appraising the color spectra must be used. Several reference systems exist and we have chosen to use the CIELAB color system when performing color match between proof and print.

The CIELAB space is a 3 dimensional uniform color space whose coordinates represent:

- "L", the luminance of a color;
- "a", the redness/greeness of a color;
- "b", the yellowness/blueness of a color.

With the aid of a colorimeter, a color can be uniquely defined and measured by these three values. Refer to the figure below which shows a 2 dimensional "slice" of the CIELAB color space where the L (luminance) value is held constant.

THE COLOR WEDGE

The Color Wedge is the sample reference pattern which is used when setting the Color Balance on the Iris. With every TRANS/4 system, a magtape containing the file COLWDG.P is received.

When this Color Wedge is printed, it appears as a pattern of 4 color strips (CMYB) with the lightness of points (luminance) varying over a range of 0 - 100% dot (or 255 - 0 when system values are used).

Set Densities on the Iris

The first step is to create a proof of the Color Wedge by outputting the COLWDG.P file to the Iris Printer. Refer to Chapter 1. for instructions on how to print files on the Iris printer.

Next, using the X-Rite 918 Colorimeter (refer to X-Rite Operation Manual), measure the luminance (L value on the X-Rite display) at 100% dot for Cyan. Write down this value on the form provided on the next page. Move the colorimeter down the scale by 5% dot to the 95% dot position. Measure the luminance and write it down. Continue measuring at 5% dot increments until you reach 50% dot. (Fewer measurements can be made at the middle range.)

Starting from 100% dot and going down, try to notice where the luminance values steadily rise. The place on the % dot scale where the luminance begins to increase consistently represents the the density value which should be set on the Iris FEP.

Repeat this procedure for Yellow, Magenta and Black

An example follows.

Baseline Density/ Luminance

% Dot	CYAN	MAGENTA	YELLOW	BLACK
100 %				
95				
90				
85				
80				
75				
70				
65				
60				
55				
50				
45				
40				
35				
30				
25				
20				
15				
10				
5				
0				

Date:__________ Name:__________

Drum Speed:______ Paper:__________

An Example

Suppose the luminances were measured for CMYB and the following form was filled in with this data.

Baseline Density/ Luminance

% Dot	CYAN	MAGENTA	YELLOW	BLACK
100 %	55.77	50.02	88.11	11.69
95	55.49	49.81	87.46	10.48
90	55.54	50.02	88.06	10.99
85	55.75	49.90	87.69	10.42
80	55.63	* 50.11	87.99	11.49
75	* 56.42	50.46	87.69	9.33
70	58.18	51.48	* 87.85	* 12.08
65				
60	60.69	53.73	88.34	17.22
55				
50	63.66	55.84	88.69	22.51
45				
40				
35				
30				
25				
20				
15				
10				
5				
0				

Find the low point, or minimum, at which the luminance values start to steadily rise. Look at our example: after some initial fluctuation, the luminance rises steadily at 75% dot for Cyan from a minimum value of 56.42. Likewise, for Magenta, the low point would be at 80% dot. For Yellow - 70% dot and for Black - 70% dot.

ENTER THE DENSITY DATA

To enter the new densities on the Iris FEP, select SET UP, **default configuration**, with CLT and HUE disabled. Then select EDIT ,**Configuration** and then **Color Balance** .

The following screen appears. Enter the new density values into the FEP:

Make sure that the Contrast field remains set to 100.

3.2 CREATING AN OPTIMAL EXCURVE

This procedure creates an Excurve which is necessary for creating Color Transforms. Steps 1-5 outlined below, need only be done once - before creating the first custom Color Transform. Afterwards, it need not be repeated. Steps 6 - 9, however, require a press system print and need to be re-done for every custom Color Transform.

Before creating an Excurve, be sure that you have read the previous section of this chapter and have already adjusted the Color Balance on the Iris printer.

To create the Excurve, follow the steps below:

1. Create a Proof

Create a new proof of the sample Color Wedge pattern or use the last proof which was produced when *Adjusting the Color Balance*. If you create a new proof, make sure that the newly adjusted densities have been entered (as described in the previous section of this chapter), before printing.

2. Set up the Colorimeter

Set-up the X-Rite colorimeter as described in the X-Rite Operation Manual. The "store data" function must be activated and cleared of all previous measurements. Select the L.a.b measurement scale. Set illumin to D50/10 deg if you are using a viewing box with a color temperature of 5000 deg K or set D65/10 deg for 6500 deg K.

3. Measure the Color Wedge Proof

The order in which you measure is imporant. Starting at 100% dot for Cyan, measure the luminance with the colorimeter. Now measure Magenta at 100% dot,Yellow at 100% dot and then Black at 100% dot. Continue measuring in this order at 5%dot intervals until you reach 0% dot (e.g. 95%, 90%, 85%, ...etc).

4. Download the Measurements

When all 84 points have been measured, you may now transfer the data to the Scitex workstation.

Connect the X-Rite colorimeter to the Whisper via the supplied cable. The cable connects to the socket on the SCD card. Verify that the X-Rite is set to Baud Rate 9600, RCI = ON, CR with no Line Feed, and X-On/X-Off enabled. (See "RS232 I/O Parameters" section of the X-Rite Operation Manual.)

Next, on the Whisper station, select **Table**, **New Table** and choose **Colorimeter**. The following screen is displayed:

Type in a new name for the measurements file leaving the colorimeter field set to X-Rite 918. Press the **Receive Measures** softkey and the data transfer will take place. To monitor the progress, look at the X-Rite display.

5. Calculate the Excurve

On the Scitex workstation, **select Table, New Table**, then choose **Auto LUT**. The following screen appears:

Enter the following parameters:

file: The new Excurve name (type LUT). This is the output of the function.

Proof Measurements file: The name of the measurements file downloaded (step 4.).

Print Measurements file: leave blank

Press Save & Exit softkey and the new Excurve will be calculated for Cyan, Magenta and Yellow.

The steps which follow are for the black separation of the Excurve.

6. Create a print of the Color Wedge on the target press system

Using the COLWDG.P file, create a print on the desired press system.

7. Measure the print

Starting at 100% dot, measure the luminance of black on the print. Write it down on the form provided on the next page. Continue to measure and note the luminance values at 5% dot intervals until you reach 0% dot.

8. Work with the Iris Proof

Begin by looking at the "Excurve Data Gathering" form and reading the luminance value that was previously measured from the print at 100% dot. Now start working on the Iris proof with the colorimeter. Locate this value (or closest value) on the last Iris Proof. Note the location on the %dot scale and enter it on the corresponding line of the form provided.

Repeat this for all the 21 luminances which have been measured on the print.

EXCURVE DATA GATHERING

BLACK % Dot	PRINT: Measured Luminance	PROOF: % dot of corresponding luminance
100 %		
95		
90		
85		
80		
75		
70		
65		
60		
55		
50		
45		
40		
35		
30		
25		
20		
15		
10		
5		
0		

Table Name__________ Date:__________ By:__________

Density (CMYB)__________ Drum Speed:________ Paper Type_____

An Example:

In the example below, the luminance measured on the print at 95% dot for black was 14.67. On the Iris proof the luminance of 14.50 was found to be the closest. This occured at the 86% dot location on the scale. The form would be entered as shown below:

9. Entering the Data for Black

To enter the data you have just determined for black, select the Excurve file (LUT file which you created in step 5) from the Library list. The following screen appears:

Press the List/Modify softkey which brings you to a screen which typically, would look like this:

For each of the 21 points, a new value for black must be entered. To do this, press the Change softkey. This version of the screen appears:

The system prompts you for the point number. The point number refers to the line # located under the "#" heading. After entering a line number, press the confirm change softkey. The values which currently exist for that line are displayed: .

You are also prompted to enter the new data. The values under "On Scanned File" should not be changed therefore skip over to the "On Scanner" column by pressing the ENTER key. Use the cursor (—>) key here to skip over to change the Black entry, leaving CMY as they appear. For black, enter the values that you filled in on the right-most column of the "Excurve Data Gathering" form ("Proof: %Dot of corresponding luminance"). The left-most column of the form ("%Dot") corresponds to the values of the "On Scanned File" entries on the screen.

To continue, press **Another** softkey and repeat the procedure until the black value for all 21 points have been modified. Once this is done, the Excurve is complete.

3.3 THE COLOR TRANSFORM

The Color Transform performs a color translation of a CT, LW or Final layout file, so that an Iris proof of this file will closely resemble the final press print.

Several Color Transforms have already been created at the factory to simulate some popular proofing and printing systems. These include: Matchprint, Cromalin, Offset, Gravure.... A magnetic tape containing the corresponding Color Transform files (type NDL) is supplied with every TRANS/4 system. Read this section to create a Color Transform for some other printing method.

Color Transforms are created by matching an Iris reference proof to a press reference proof (hereafter referred to as the "print"). The print is obtained by printing the color separations on the press with the same ink, paper and speed as would be used on the final job.

In this section, we will discuss ways to :

- Create a Color Transform
- Evaluate a Color Transform
- Edit a Color Transform
- Manipulate a Color Transform
- Apply a Color Transform to a Page

3.3.1 Creating a Custom Color Transform

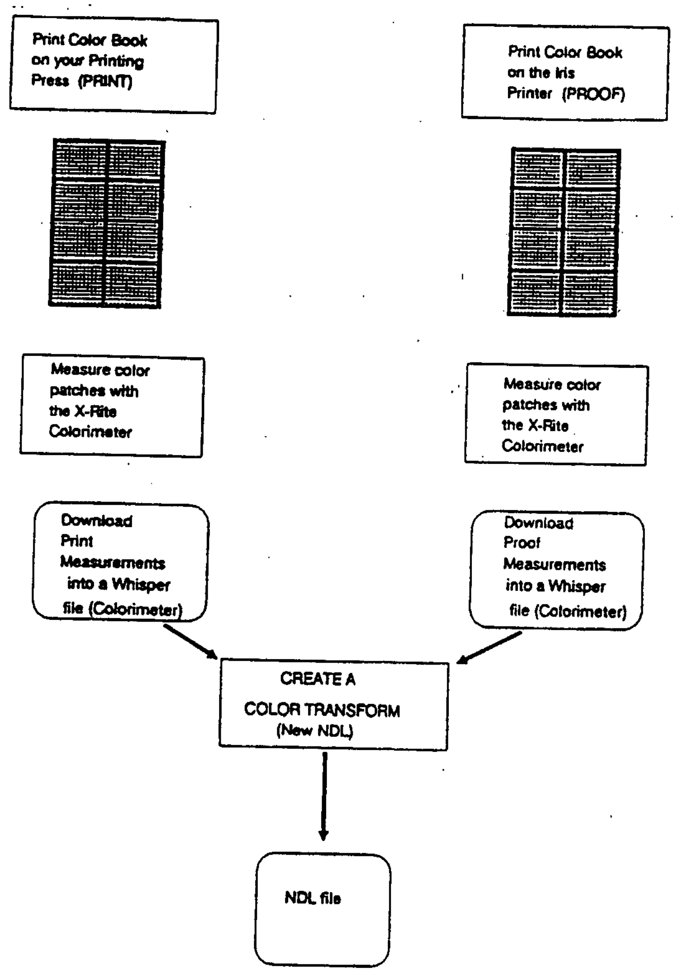
The inputs to the Color Transform Function are:

- a set of colorimeter measurements taken from a sample pattern that was printed on the Iris printer (referred to as "proof").
- a set of colorimeter measurements taken from the same sample pattern that was printed on the desired press (referred to as "print").

The output of the Color Transform function is:

- a special Color Transform file (type NDL) which, when applied to a CT, LW or Final Layout, produces a color match of the print.

CREATING A COLOR TRANSFORM



To create a Color Transform file, follow the steps below:

1. Create a Print and Proof

The first step is to create a proof of the sample COLOR BOOK pattern which has been supplied with TRANS/4. Output the COLBOOK.P file to the Iris Printer. (Refer to Chapter 1. for instructions on how to print files to the Iris Printer.) Next, create a print of COLBOOK.P by producing 4 color separations and duplicating the final printing process.

2. Set up the Colorimeter

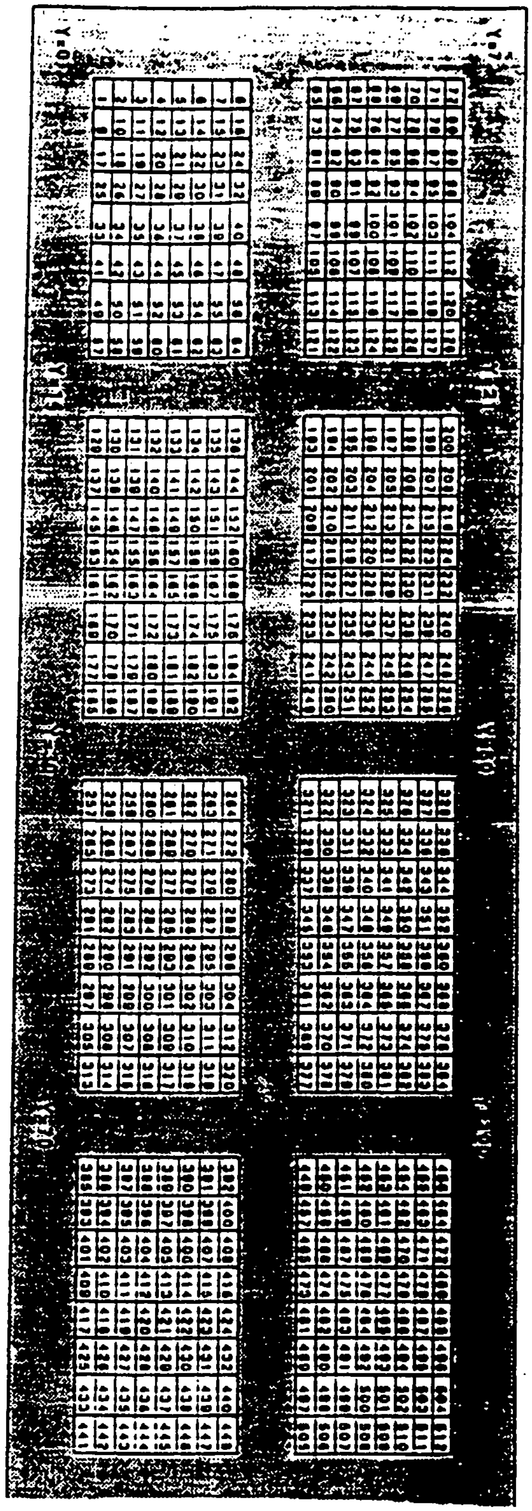
Set-up the X-Rite colorimeter model 918. The "store data" feature must be activated and cleared of all previous measurements. Select the L.a.b. measurement scale. Set Illumin to D50/10 deg if you are using a viewing box with a color temperature of 5000 deg K or set D65/10 deg for 6500 deg K.

3. Measure the Print

The COLOR BOOK pattern consists of a grid of 8 boxes. Each box has a different constant value for yellow and contains 64 patches. Measure the C.I.E L*a*b* of each patch on the print with the colorimeter. Note that this will require 512 measurements. The order in which you measure the patches is very important. Proceed as follows:

- First measure the white patch. This corresponds to patch #1 marked on the figure above.

- Continue measuring in the order in which the patches are numbered: 2, 3, 4, ...etc. Measure all 64 patches in the box and then proceed to measure the patches in the next box on the right (where Y = 7). Continue this until the darkest patch (#512) is measured. If at any point during the measuring, you need to stop, you can easily return to the point where you left off. Read in the last patch number read by the colorimeter by pressing the ILLUM and DIF keys simultaneiously on the X-Rite. The measurement number is displayed under the SEND DATA ? message. Select "No" for the queries which follow and then proceed to read the next patch. Refer to the figure on the next page to locate the position from which to continue measuring.



COORDINATES SCITEX TESTABLE

4. Download Print Measurements From X-Rite To Whisper

When all 512 patches have been measured, you may now transfer the data to the Scitex workstation.

Connect the X-Rite colorimeter to the Whisper via the supplied cable. The cable connects to the socket on the SCD card. Verify that the X-Rite is set to Baud Rate 9600, RCI = ON, CR with no Line Feed and X-On/X-Off enabled. (Refer to "RS232 I/O Parameters" of the X-Rite Operations Manual.)

Next, on the Whisper, select **Table, New Table** and choose **Colorimeter**. The following screen appears.

Type in a new name for the print measurements file leaving the colorimeter field set to X-Rite 918. Press the **Receive Measures** softkey and the data transfer will take place. To monitor the progress, look at the X-Rite display. There will be some delay and then the display will start flashing.

5. Measure the Proof

Repeat the same procedure for the proof as was described in Step 3."Measure the Print."

» **Note: When creating a Color Transform for a different printing method, Steps 5 and 6 need not be repeated since this data will not change.**

6. Download Proof Measurements From X-Rite To Whisper

Download the proof data from the X-Rite to the Whisper exactly as was described in Step 4. for the print. Note, however, that the name you assign for the measurements file should be different.

7. Calculate the Color Transform

On the Scitex workstation, select **Table, New Table,** then choose **New NDL** The following screen appears:

When calculating a particular Color Transform **for the first time,** enter the following parameters:

File: Enter a new Color Transform file name
Color Transform file name : LEAVE IT BLANK. (Used for Pass 2)
Proof measurements file : Enter the name as in step 4.
Print measurements file : Enter the name as in step 6.

Press the **Save & Exit** softkey to start the calculation.

Noitce that the *Color Transform file* name has been left blank. This field is reserved for use when calculating the Color Transform on the 2nd Pass. The 2nd Pass calculation is optional and can be used to produce a Color Transform with possibly closer color match results. The procedure is outlined as follows:

PASS 1:

- Perform Steps 1 - 6 as described in the previous pages.
- Perform step 7. - Calculate the Color Transform - leaving the *Color Transform file* field blank.
- Press the **Save & Exit** softkey to perform the calculation

PASS 2 (optional):

- Create an Iris Proof of the Color Book, while applying the Color Transform file created in Pass 1 above.
- Measure this proof with the colorimeter (as described in step 5.).
- Download the proof measurements into a new proof measurements file (as desribed in step 6).
- Calculate the new Color Transform (step 7) by specifying a new proof measurements file. Also, fill in the *Color Transform file* field with the NDL file name which was just created in Pass 1.
- Press the **Save & Exit** softkey to perform the 2nd Pass calculation.

In both passes, the **Save & Exit** softkey is pressed to initiate the calculation. To display the progress indicators, select **show** and then **Bkgd status.** It is then possible to observe the messages indicating the completion of the various stages of the calculation by noting the progress indicator bar graph. Three different bar graphs will appear during the procedure. If the Color Transform has been successfully created and no modifications necessary, then back up this Color Transform file onto magnetic tape.

To quit the calculation at any point, select Abort located under the bar graph.

3.3.2 Evaluating a Color Transform

After you have completed the Color Transform calculation, you can evaluate the results by applying the Color Transform to the Color Book CT and printing it on the Iris Printer once again. You can then inspect the proof visually or with the aid of a colorimeter.

An alternative approach is to let the TRANS/4 system analyze the results quantitatively. First, create a new proof with the Color Transform applied to the Color Book CT. Using the colorimeter, measure the 512 color patches and download them into a Proof measurements file. Select New NDL (see page 3-27) and fill in any bogus name for the file prompt. (Actually, no new file is created.) Enter the new Proof Measurements file name and the original Print Measurements file name. Pressing the **Analyze** softkey will display a special screen which gives you an idea of how close your "transformed" proof matches your print. A typical screen may appear as follows:

The units which appear under "D LAB" represent the "size" of the error. The 0 - 1 row represents the smallest possible error. The 1 - 2 row represents errors of a slightly greater magnitude, however they are still undetectable by the human eye. As you continue down the rows, these numbers increase and represent larger and larger errors. The dashed lines which appear next to the error scales represent the number of patches which were found to fall within that error range. Note that for display purposes, the maximum number of errors per error level has been normalized to 10 on this graph.

When most of the errors fall within the 0 - 1, 1 - 2, 2 - 3 or 3 - 4 range, the Color Transform that has been created is quite good. An overall average rating is given as well as the error of greatest magnitude which has been detected.

A large size error (anything greater than 7 - 8) will most likely be due to a bad colorimeter measurement or perhaps due to a non-printable color.

The system provides you with the actual CMYK position of the Color Book for the greatest error detected. This is in the event you wish to edit this particularly troublesome patch.

3.3.3 Editing a Color Transform

The last Color Transform created in the system may be edited. This might be done in order to adjust CMY points which have not been successfully transformed for one reason or another. (Note: Black may not be edited.) To access this function select **New NDL** (see page 3-27). Enter a new NDL file name leaving the other fields blank. Now, press the **Edit** softkey to begin editing.

Start by specifying the units you are working with by toggling with the space bar between " % dot" or "system values". Next, enter the values for C,M,Y,B of the patch to be edited alongside *Near* located at the bottom left portion of the screen. *Note that only points which have been measured with the colorimeter may be edited.* When entering *Near:* values, the TRANS/4 system determines the closest location of the point actually measured and displays it alongside the *At* : row.

Once the closest measured position has been determined, the value of the Color Transform at that position is displayed alongside *Before Edit*. Now, key-in the new values alongside the *After Edit* position. Select *OK* when you are satisfied with your entries by pressing the ENTER key. If you wish to modify any of the keyed-in data, press the SPACE BAR to clear the *OK* and re-enter the data. How to edit points to obtain desired proofing results is a matter of experience as well as trial-and-error and beyond the scope of this manual.

You may continue to make further modifications. As the points are edited they are logged and displayed to the top portion of the screen. The left top portion shows the position of the CMYB point edited and the right top portion contains the values before (*old*) and after (*new*) editing.

When you have completed all modifications, make a back-up copy of the Color Transform file onto magnetic tape.

3.3.4 Manipulating the Color Transform

Once the Color Transform file (NDL) as been created, there are essentially three possible ways to manipulate it:

- Adding Black to an existing Color Transform
- Converting a Color Transform file to a CT
- Converting a CT back to a Color Transform file

If any changes have been made to the original Color Transform, it is important to remember to back it up on magnetic tape.

Adding Black To An Existing Color Transform

Currently, when creating a Color Transform using the Color Book supplied, the value for black is not transformed. You may optionally add the black transform via the **Add Black** function. The Color Transform created when black is introduced is derived from the existing NDL file and not measured off a Color Book. .To bring up the screen, select **Table, New Table** and choose **Add Black.**

For the *file* field, enter the NDL file name which currently exists (without black). For the *New NDL* field, enter the name of the new Color Transform file which will now include different values of black.

Converting a Color Transform to a CT

The advantage to converting an NDL file to a CT file is that once you have a CT you can apply the data correction function that exists for CT's on the Scitex workstation. That is, you can apply a one-dimensional gradation onto the CT. For example, if you wanted to raise Cyan middletone by 5%, you would be able to accomplish this easily on the CT file. This is opposed to editing 512 points on the NDL Color Transform file.

After some change has been made to the CT file, it may be converted back to an NDL Color Transform file. To convert a Color Transform file to a CT, select Start work on an existing NDL file and the following appears:

The NDL file name which was previously selected will be displayed.

For the *CT File Name* field, enter a new name.

Notes:

Be sure to use the **Add Black** function first if you wish to produce a black separation in the CT.

When creating a CT file, the %dot values which exist in the Color Transform file are compressed into a smaller range. Therefore, the correspondence between %dot values between NDL (actual) and CT (displayed) files is not one-to-one. The graph below may be used to convert the displayed %Dot values into actual %Dot values. For example, a displayed value of 20% dot has an actual % dot value of 15% dot in the NDL file.

Converting a CT back to a Color Transform

If a Color Transform file has been converted to a CT and some editing has been done, the file must be brought back to its original form (NDL) in order to effect any changes on proofing.

Only files which have been *previously converted* to CT files, via the CT<— NDL function, may be brought back to the Color Transform format. Select **Table, New Table** and choose CT —> NDL to bring up this screen:

For the *File* field, enter a new Color Transform (NDL) file name. For the *CT File Name* field, enter the CT file you wish to convert.

3.3.5 Applying a Color Transform to a Page

Once the Color Transform has been set up and finalized it can be applied to a particular page for it to have its effect. Recall that your system can be calibrated to have any number of different Color Transforms which represent different printing environments.

To apply a Color Transform to a file, select the **Proof** Function. (This function is described in its entirety in Chapter 1. TRANS/4 Operation.) The inputs to **Proof** are the CT, LW or Final Layout file to be printed as well as the Excurve and Color Transform file you wish to apply to it. The output is a "transformed" image which is printed directly on the Iris printer.

Appendix I - HSP Installation Procedure

Introduction

The High Speed Processor (HSP) is a high speed co-processor for Whisper based Response systems. It functions as an accelerator for the Response CPU, increasing the speed and efficiency of the system. The HSP operates with all Whisper based Response and Micro-Response configurations.

The HSP Kit

The HSP (Catalog # 510K24087) is supplied as a kit containing the following:

HSP Hardware Installation

Before beginning the installation procedure, operate the system to verify that it is in working order.

Note: The HSP can only be installed in systems operating on software version 4.2 and up.

Perform the following to install the HSP kit in your Whisper based Response system.

1. Power-down the system.
2. Remove the Whisper cabinet front panel.
3. Verify that the DIP switches on the HSPI and PM2 boards are set as follows:

4. Insert the HSPA, HSPI and PM2 boards into the slots appropriate for your Whisper based Response (see Table 1.) or Micro-Response (see Table 2).

Note: In STARTER workstations, slot 2 must be free. If the MLT board is in slot 2, remove and insert in slot 9.

5. Connect the jumper boards between the following HSP boards :
   - HSPA J1 and HSPI J1
   - HSPA J3 and HSPI J3
   - HSPI J2 and PM2 J2

6. Replace the Whisper cabinet front panel.
7. Power-on the System.

HSP Software Configuration

In the configuration process we inform the system that the HSP boards have been installed. Perform the following from the Response application to configure the HSP boards in the system.

- Select **SERVICES** from the Utility menu.
- Select **CONFIGURATION** from the sub-menu.
- Select **STATION CONFIG** from this screen.
- Select **BOARD CONFIG** from this screen to display the following:

Toggle the HSP and PM2 fields to display **YES** with the Space Bar.

Select **SAVE & EXIT**.

Reboot the system.

HSP Testing

Test the HSP by performing the board tests through the on-line diagnostic handler and then test the HSP application by performing some Response operations.

HSP Board Diagnostics

Perform the following procedure to test the HSP and PM2 boards with the on-line diagnostic handler.

1. Invoke the diagnostic handler.
   - Select **SERVICES** from the utility menu on the Response.
   - Select **DIAGNOSTICS** from the Services Menu.

The startup screen is displayed on the terminal as shown below:

2. Test the HSPI and HSPA boards
   - Enter U (CR) for UNIFIED
   - Enter (CR) for CHECK CONFIG.

The Unified subsystems that can be tested are displayed as shown in the figure below:

Note: Response systems have different configurations. This menu will vary depending on the configuration.

- Select HSPM-2 from this menu (enter 5 (CR) to get:

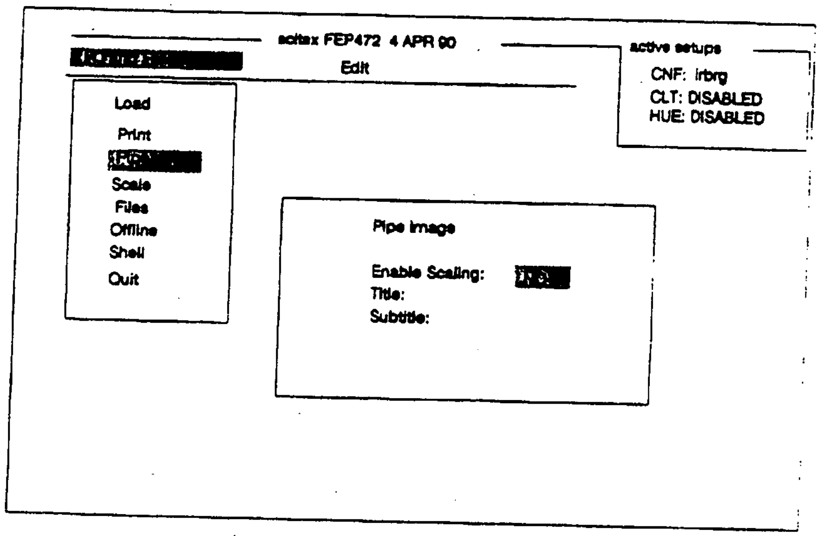
- Enter // (CR) to run all the test in a batch.

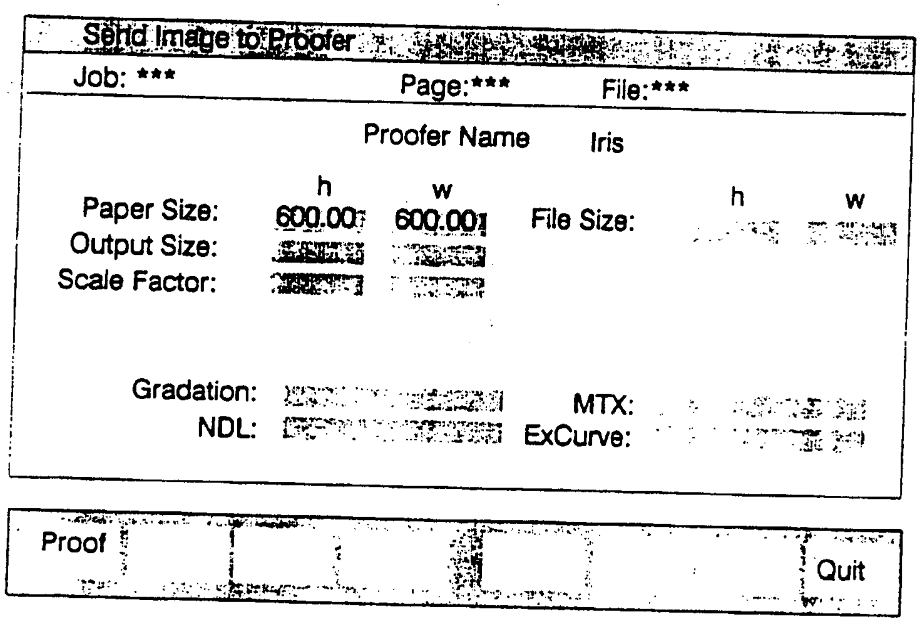
3. Test the PM2 board
   - Enter B to go back to the Unified subsystems menu
   - Select FRAME BUFFER from this menu.
   - Select PICTURE MEM from the Frame Buffer menu
   - Select PM2
   - Enter // (CR) to run all the tests in a batch.

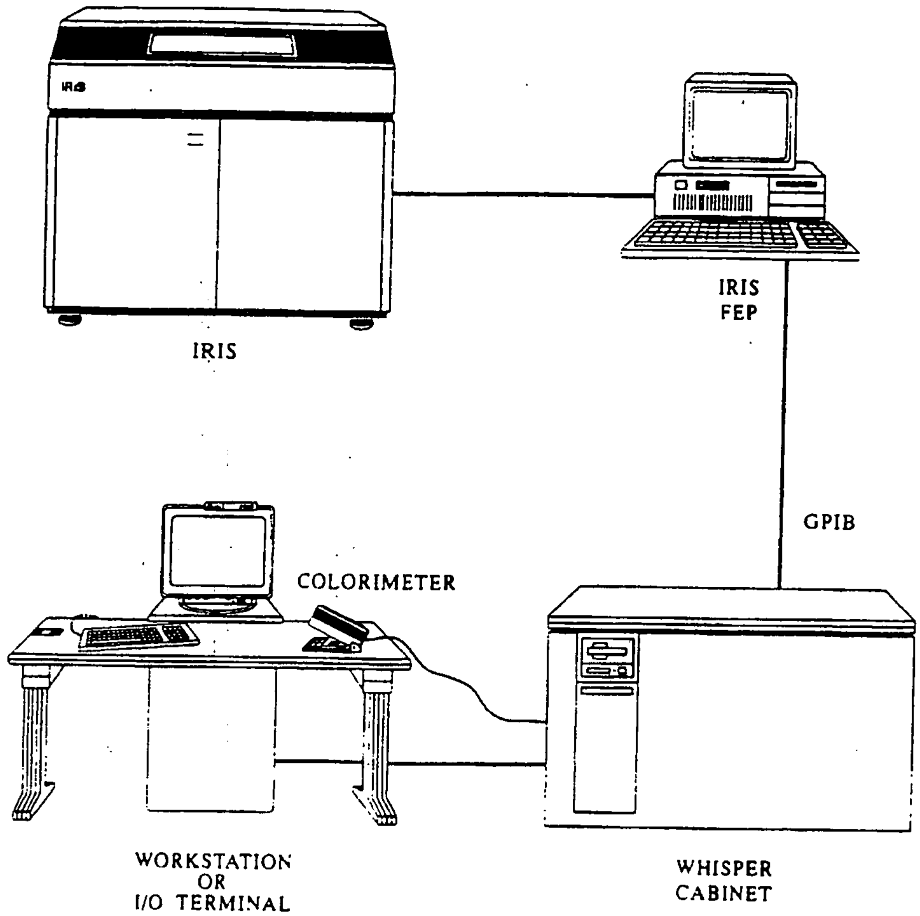
4. Quit the Diagnostic Handler by entering Q (CR). The system exits the diagnostic mode and reboots.

Application Tests

Create a LW and make sure that the process ends successfully.
Rotate a CT and make sure that the process ends successfully.


scitex FEP472 4 APR 90
Edit
active setups
CNF: irbrg
CLT: DISABLED
HUE: DISABLED
Load
Print
Scale
Files
Offline
Shell
Quit
Pipe image
Enable Scaling:
Title:
Subtitle:


Send Image to Proofer
Job: ***
Page:***
File:***
Proofer Name
Iris
h
w
Paper Size:
600.00
600.00
File Size:
h
w
Output Size:
Scale Factor:
Gradation:
MTX:
NDL:
ExCurve:
Proof
Quit



System Configuration

Workstation Type I/O

Station Name CPU5

Network Yes

Workstation Types:
I/O, Xcelerator, Assembler, Keyline, Smartview
Softproof, Rightouch, Assembler Plus, Prisma

Save & Exit	Station Config	Storage Device	SciNet	Logical Names	MIS	Secur. Info	Quit

Security Information

Workstation Type ~Assembler

Board S/N [redacted]

Password [redacted]

Option	Open	Option	Open	Option	Open
Netway	~NO	USFM	~NO	DDES	~NO
HandShake	~YES	Optic	~NO	MIS	~NO
Visionary	~NO	3M	~NO	TRANS/4	~YES
Text	~YES				

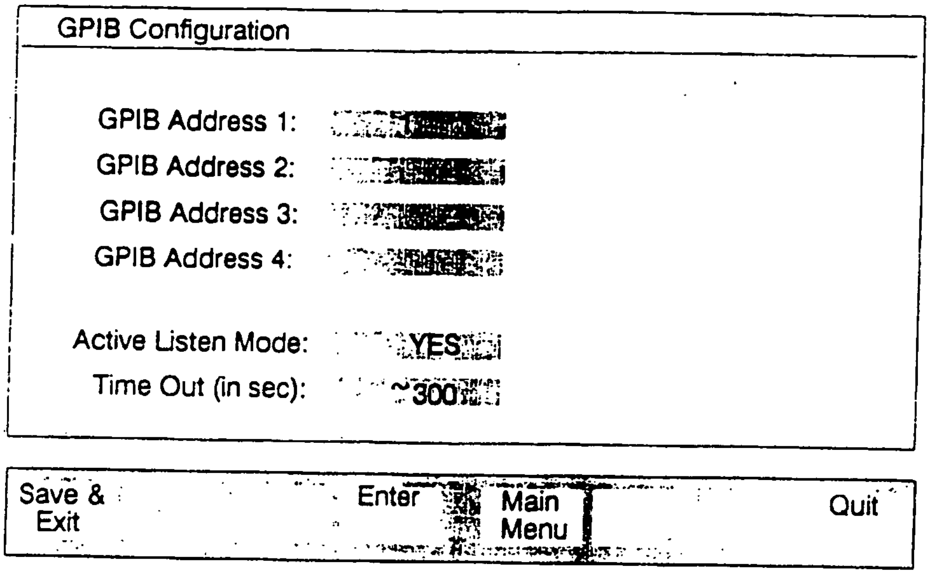
Customer # 10

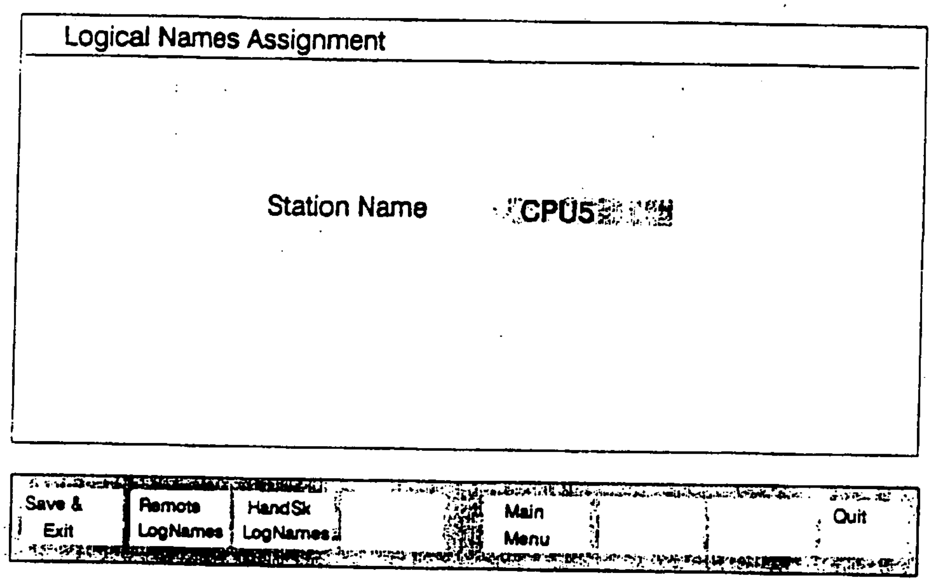
Save & Exit			Main Menu		Quit

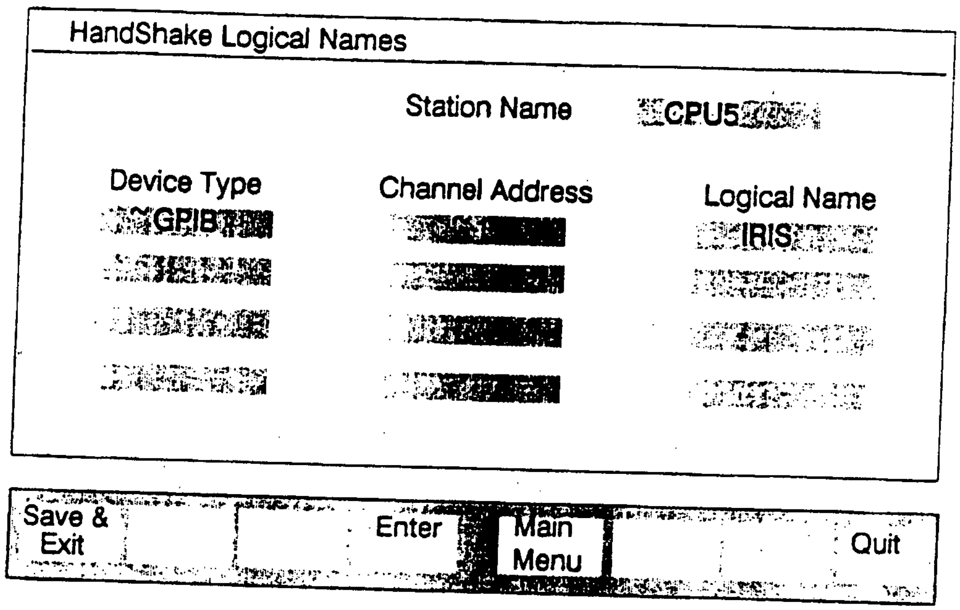
HandShake Configuration

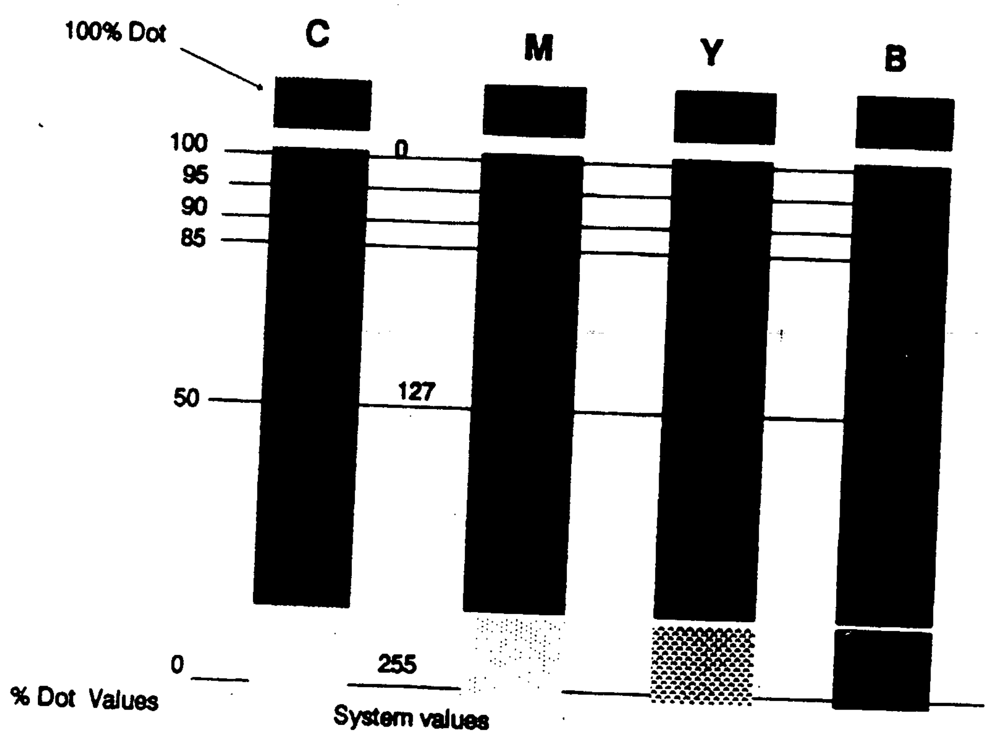
Station Name CPU5

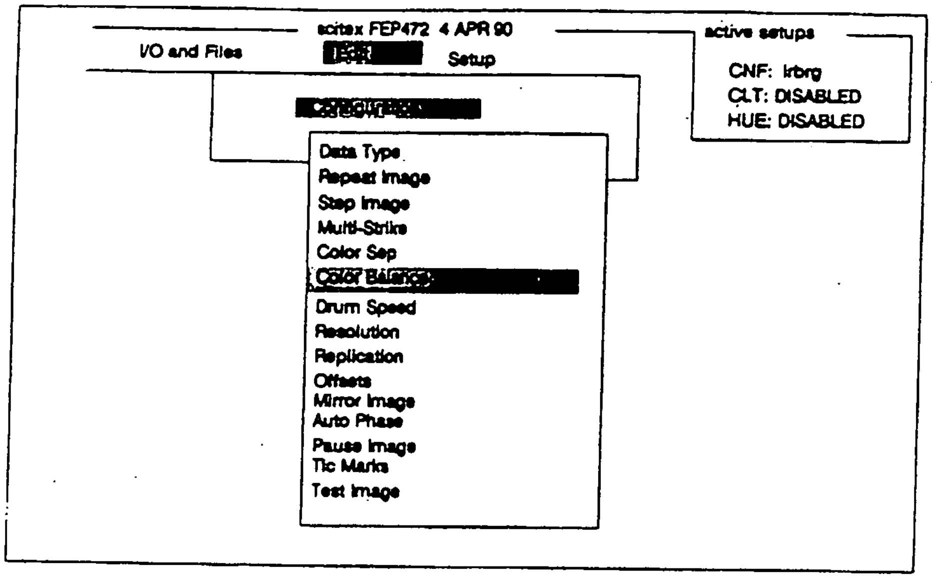
Save & Exit	RS232	GPIB	MT	Main Menu	STND	Quit

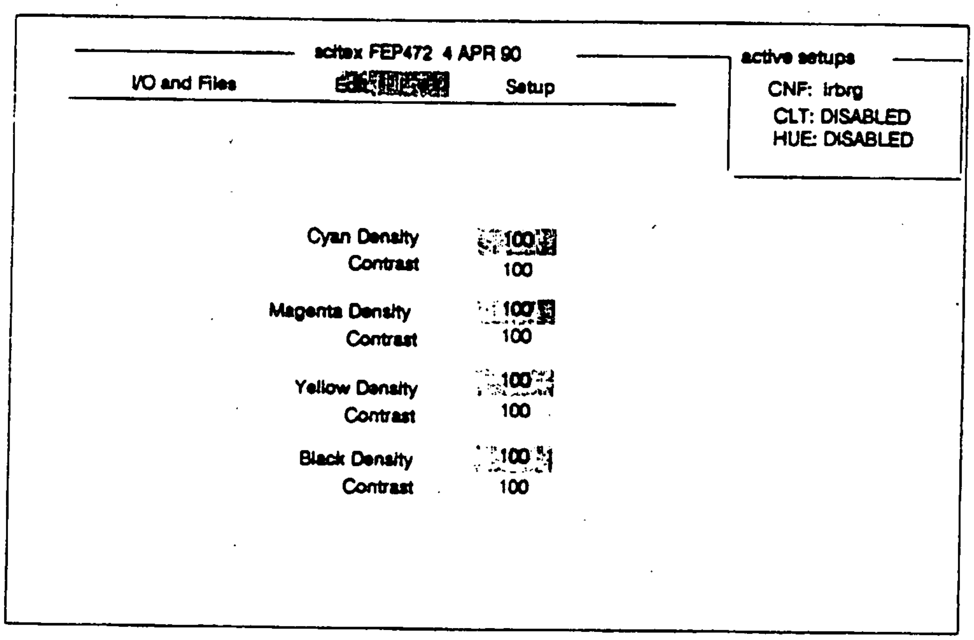

GPIB Configuration
GPIB Address 1:
GPIB Address 2:
GPIB Address 3:
GPIB Address 4:
Active Listen Mode: YES
Time Out (in sec): 300
Save & Exit
Enter
Main Menu
Quit

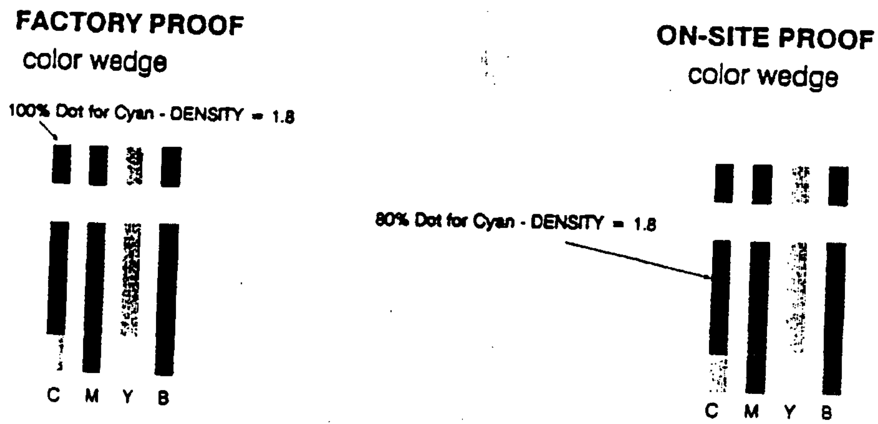

Logical Names Assignment
Station Name CPU5
Save & Exit
Remote LogNames
HandSk LogNames
Main Menu
Quit

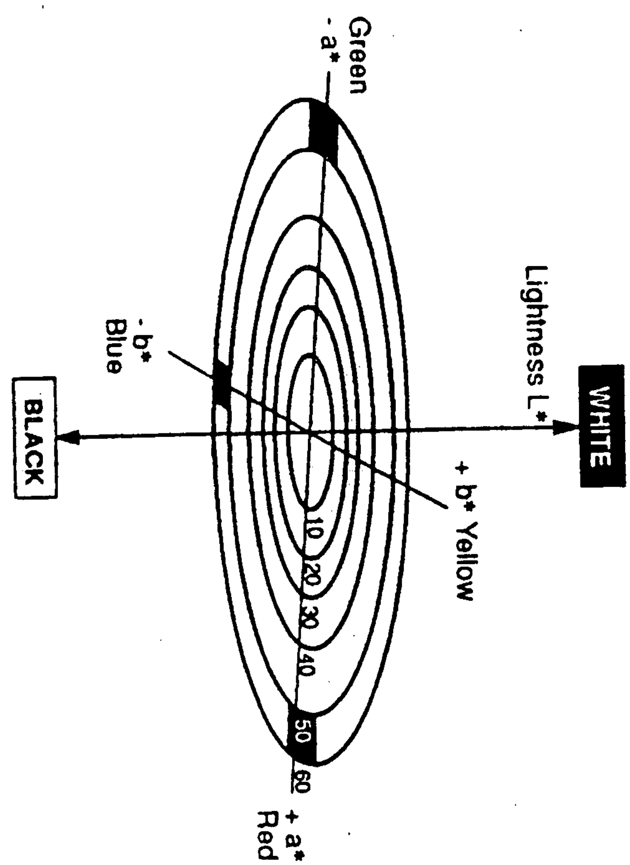

HandShake Logical Names
Station Name
CPU5
Device Type
GPIB
Channel Address
Logical Name
IRIS
Save & Exit
Enter
Main Menu
Quit

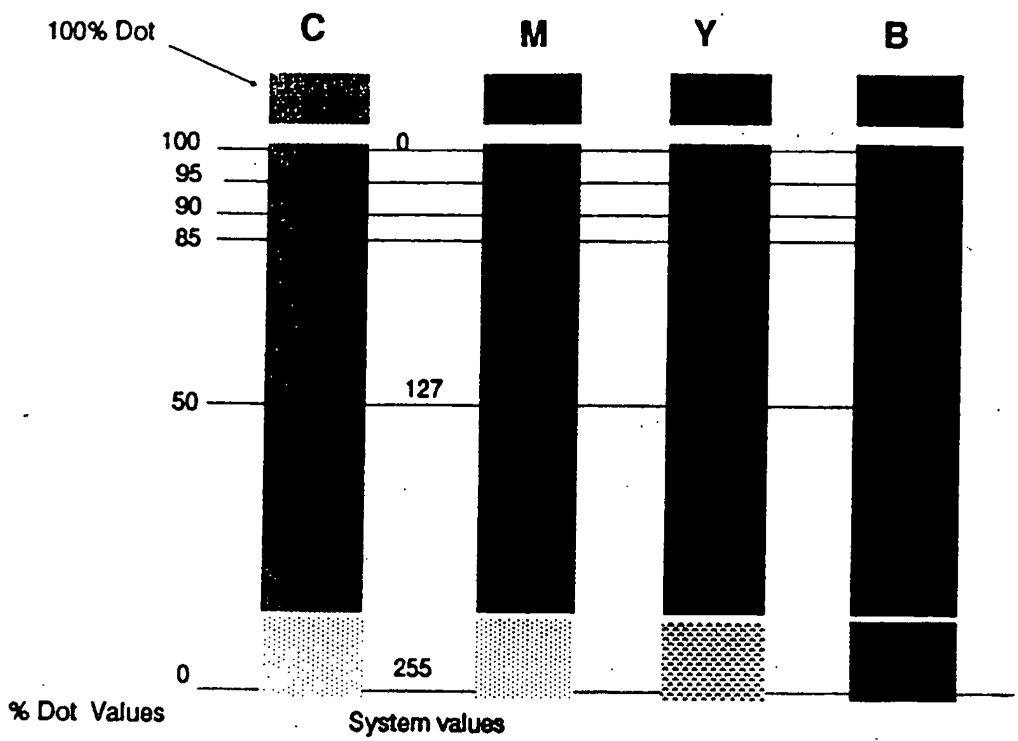

100% Dot
C
M
Y
B
100
95
90
85
50
0
0
127
255
% Dot Values
System values


scitex FEP472 4 APR 90
I/O and Files
Setup
active setups
CNF: irbrg
CLT: DISABLED
HUE: DISABLED
Data Type
Repeat Image
Step Image
Multi-Strike
Color Sep
Color Balance
Drum Speed
Resolution
Replication
Offsets
Mirror Image
Auto Phase
Pause Image
Tic Marks
Test Image


scitex FEP472 4 APR 90
I/O and Files
Setup
active setups
CNF: irbrg
CLT: DISABLED
HUE: DISABLED
Cyan Density 100
Contrast 100
Magenta Density 100
Contrast 100
Yellow Density 100
Contrast 100
Black Density 100
Contrast 100


FACTORY PROOF
color wedge
100% Dot for Cyan - DENSITY = 1.8
C M Y B
ON-SITE PROOF
color wedge
80% Dot for Cyan - DENSITY = 1.8
C M Y B


Green
- a*
- b*
Blue
BLACK
WHITE
Lightness L*
+ b* Yellow
10
20
30
40
50
60
+ a*
Red

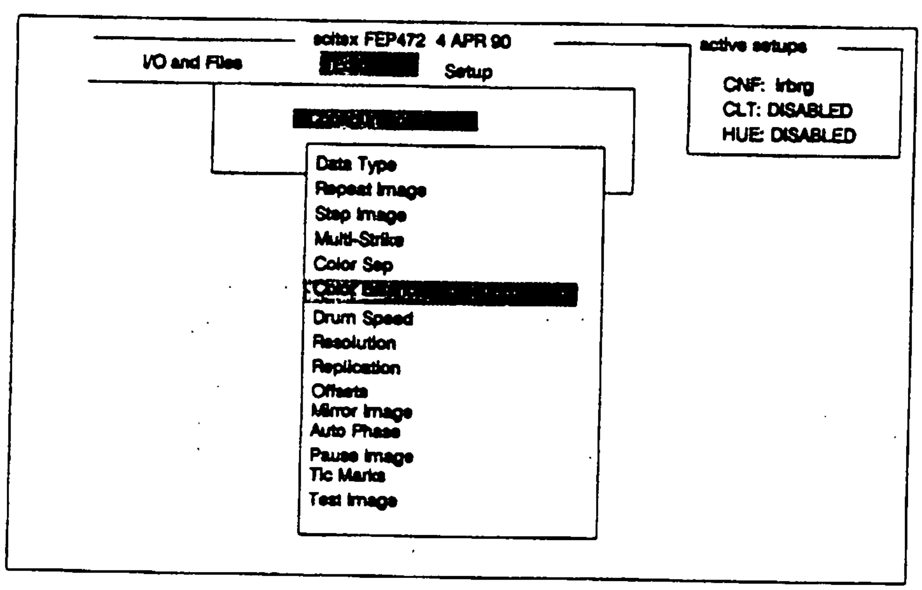

100% Dot
C
M
Y
B
100
95
90
85
50
0
0
127
255
% Dot Values
System values

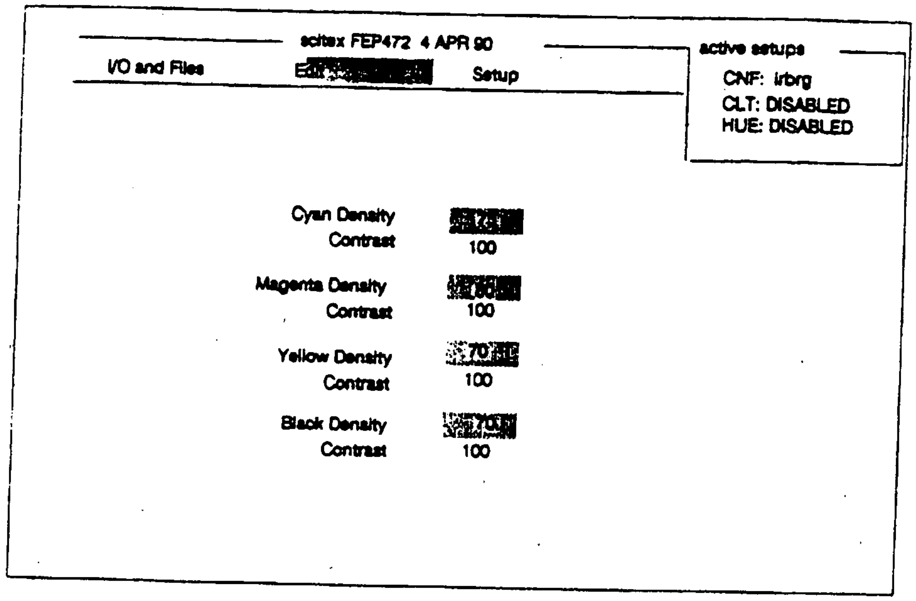
Measured Luminance

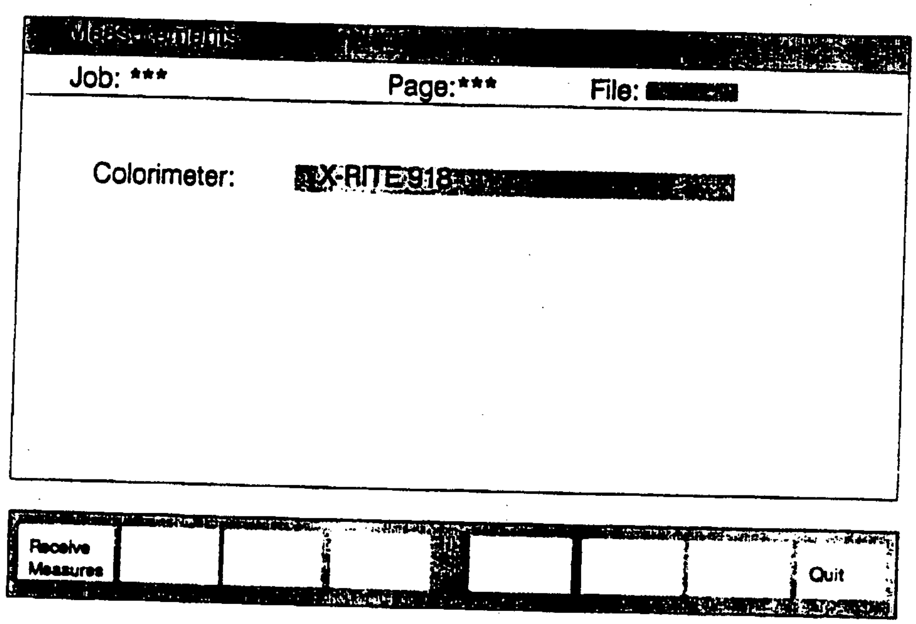
% Dot	CYAN	MAGENTA	YELLOW	BLACK
100 %				
95				
90				
85				
80				
75				
70				
65				
60				
55				
50				
45				
40				
35				
30				
25				
20				
15				
10				
5				
0				

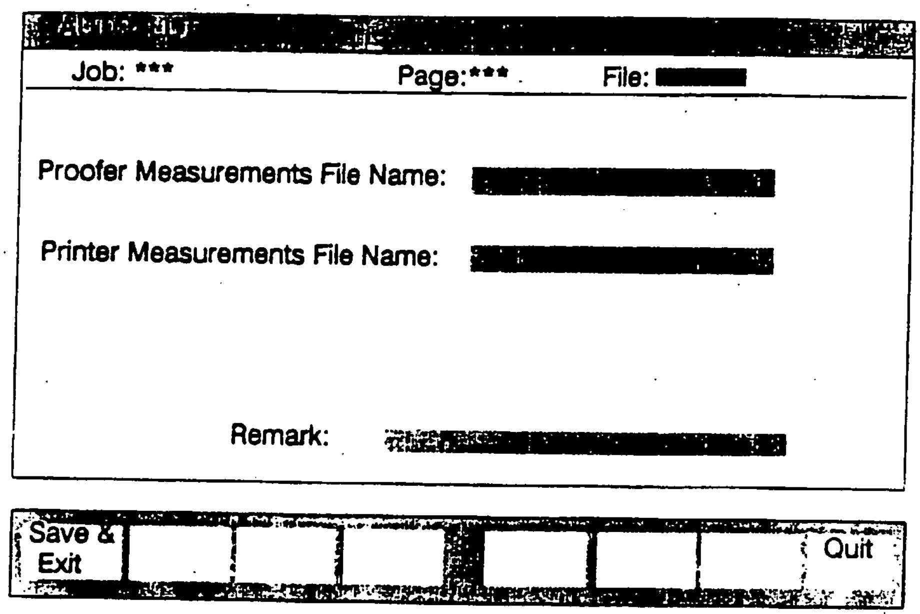
Baseline Density/ Luminance

% Dot	CYAN	MAGENTA	YELLOW	BLACK
100 %	55.77	50.02	88.11	11.69
95	55.49	49.81	87.46	10.48
90	55.54	50.02	88.06	10.99
85	55.75	49.90	87.69	10.42
80	55.63	* 50.11	87.99	11.49
75	* 56.42	50.46	87.69	9.33
70	58.18	51.48	* 87.85	* 12.08
65				
60	60.69	53.73	88.34	17.22
55				
50	63:66	55.84	88.69	22.51
45				
40				
35				
30				
25				
20				
15				
10				
5				
0				



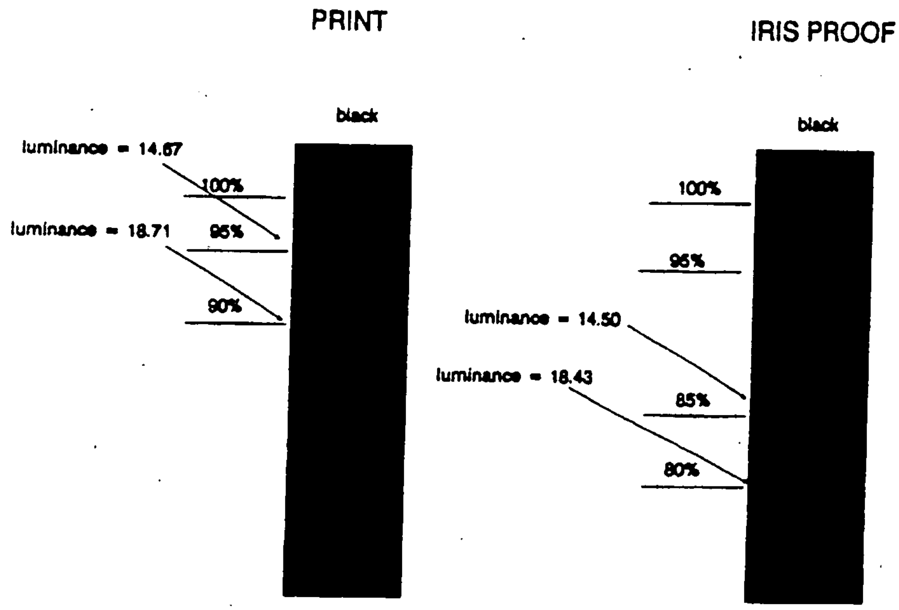

scitex FEP472 4 APR 90
I/O and Files
Setup
active setups
CNF: lrbrg
CLT: DISABLED
HUE: DISABLED
Cyan Density
Contrast
100
Magenta Density
Contrast
100
Yellow Density
70
Contrast
100
Black Density
70
Contrast
100


Job: ***
Page:***
File:
Colorimeter:
X-RITE 918
Receive Measures
Quit


Job: ***
Page:***
File:
Proofer Measurements File Name:
Printer Measurements File Name:
Remark:
Save & Exit
Quit

EXCURVE DATA GATHERING

BLACK % Dot	PRINT: Measured Luminance	PROOF: % dot of corresponding luminance
100 %		
95		
90		
85		
80		
75		
70		
65		
60		
55		
50		
45		
40		
35		
30		
25		
20		
15		
10		
5		
0		



EXCURVE DATA GATHERING

BLACK % Dot	PRINT: Measured Luminance	PROOF: % dot of corresponding luminance
100 %	11.69	100
95	10.48	85
90	10.99	80
85		
80		
75		

CREATE TABLE

JOB: j PAGE: j FILE: xyzlut

Modify Old File

Scanned File

Scanner Values

Save & Exit | Examine | List/ Modify | Abort

CREATE TABLE

JOB: j PAGE: j FILE: xyzlut

Modify List of Points

#	On Scanned File C	M	Y	B	On Scanner C	M	Y	B
1.	100.0	100.0	100.0	100.0	[illegible]	[illegible]	88.5	14.0
2.	95.0	95.0	95.0	95.0	[illegible]	[illegible]	88.8	14.6
3.	90.0	90.0	90.0	90.0	[illegible]	[illegible]	89.1	18.7
4.	85.0	85.0	85.0	85.0	[illegible]	[illegible]	89.4	25.8
5.	80.0	80.0	80.0	80.0	[illegible]	[illegible]	89.7	30.8
6.	75.0	75.0	75.0	75.0	[illegible]	[illegible]	90.0	36.2
7.	70.0	70.0	70.0	70.0	[illegible]	[illegible]	90.4	40.3
8.	65.0	65.0	65.0	65.0	[illegible]	[illegible]	90.6	44.3

Exit | Add | Change | Delete | Continue

CREATE TABLE

JOB: j | PAGE: j | FILE: xyzlut

Modify List of Points

#	On Scanned File C	M	Y	B	On Scanner C	M	Y	B
1.	[illegible]	[illegible]	[illegible]	100.0	[illegible]	[illegible]	[illegible]	[illegible]
2.	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
3.	[illegible]	[illegible]	[illegible]	90.0	[illegible]	[illegible]	[illegible]	[illegible]
4.	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
5.	[illegible]	[illegible]	[illegible]	80.0	[illegible]	[illegible]	[illegible]	[illegible]
6.	75.0	75.0	75.0	75.0	66.3	61.5	90.0	[illegible]
7.	70.0	70.0	70.0	70.0	68.2	63.7	90.4	[illegible]
8.	[illegible]	[illegible]	65.0	65.0	70.1	65.9	90.6	[illegible]

CHANGE PT. #

Cancel | Confirm Change

CREATE TABLE

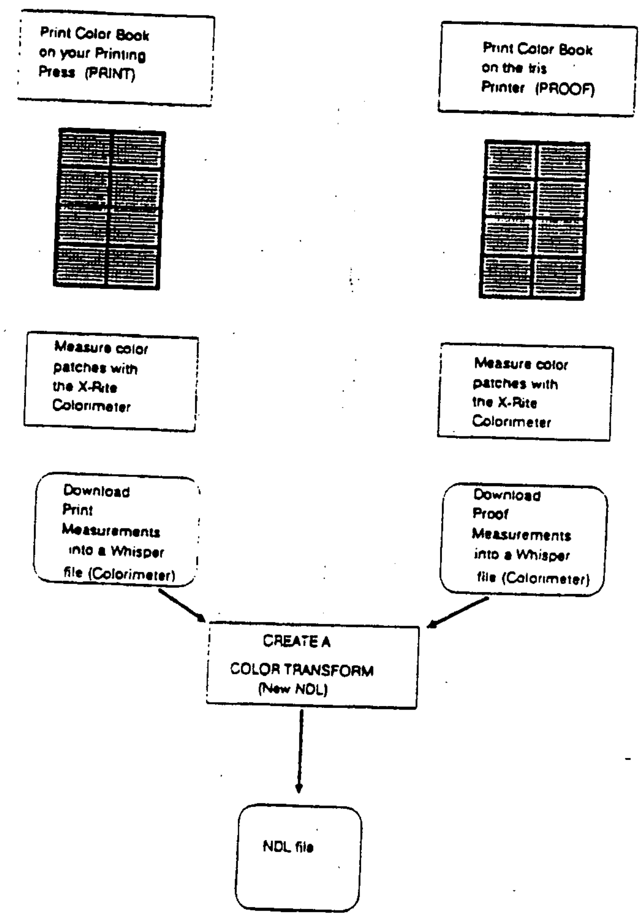
JOB: j | PAGE: j | FILE: xyzlut

Change a Point
Point 6

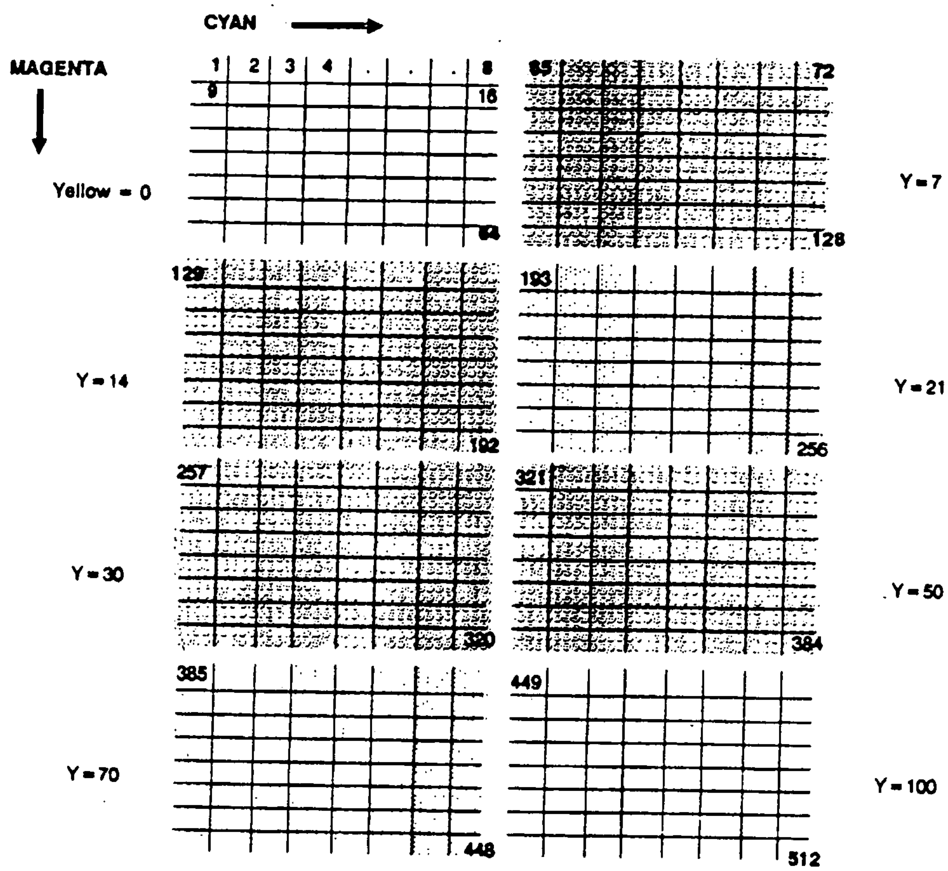
	On Scanned File C	M	Y	B	On Scanner C	M	Y	B
OLD VALUES:	75.0	75.0	75.0	75.0 %	66.3	61.5	90.0	36.2 %
NEW VALUES:	75.0	75.0	75.0	75.0 %	66.1	61.5	90.0	48.0 %

Exit | List/ Modify | Another

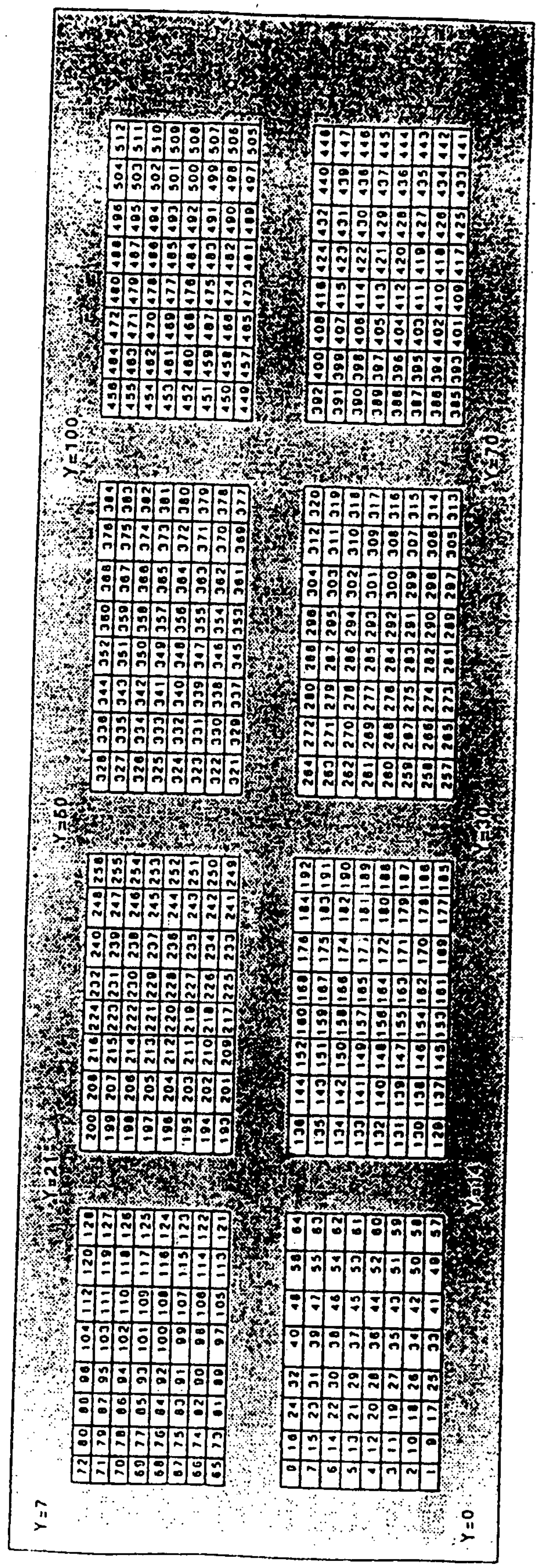
CREATING A COLOR TRANSFORM

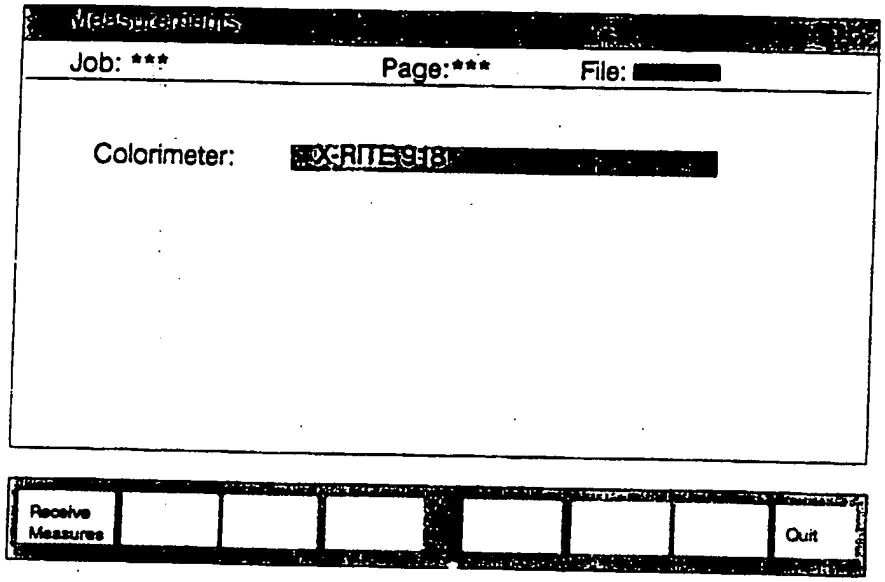


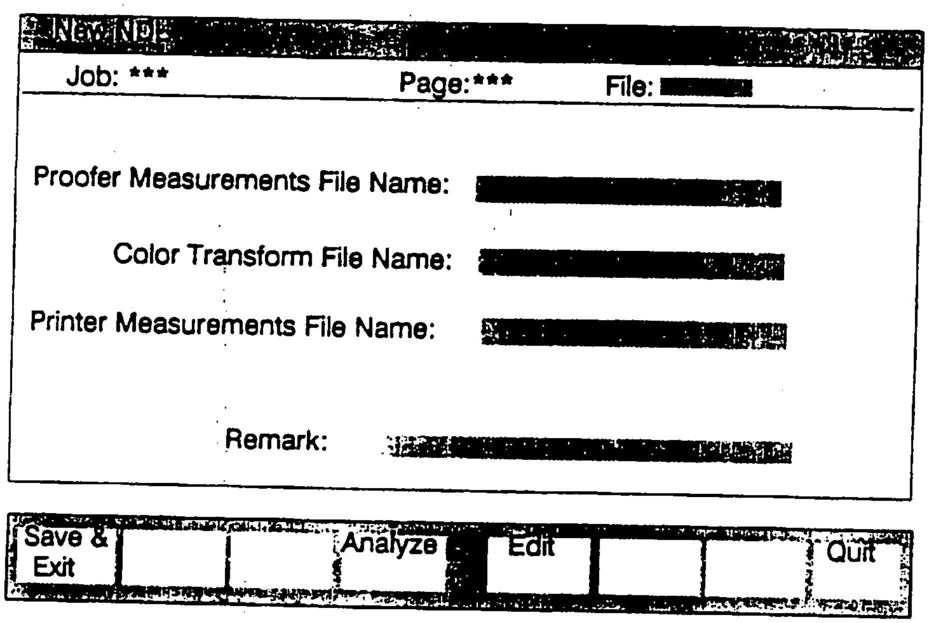
COLBOOK.P

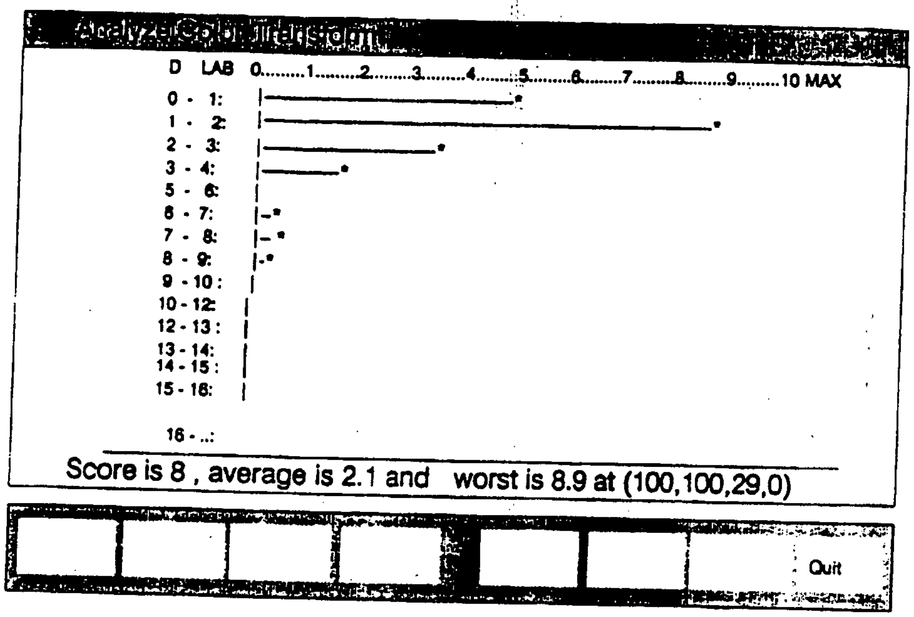

CYAN
MAGENTA
1
2
3
4
8
9
16
64
Yellow = 0
65
72
128
Y = 7
129
192
Y = 14
193
256
Y = 21
257
320
Y = 30
321
384
Y = 50
385
448
Y = 70
449
512
Y = 100

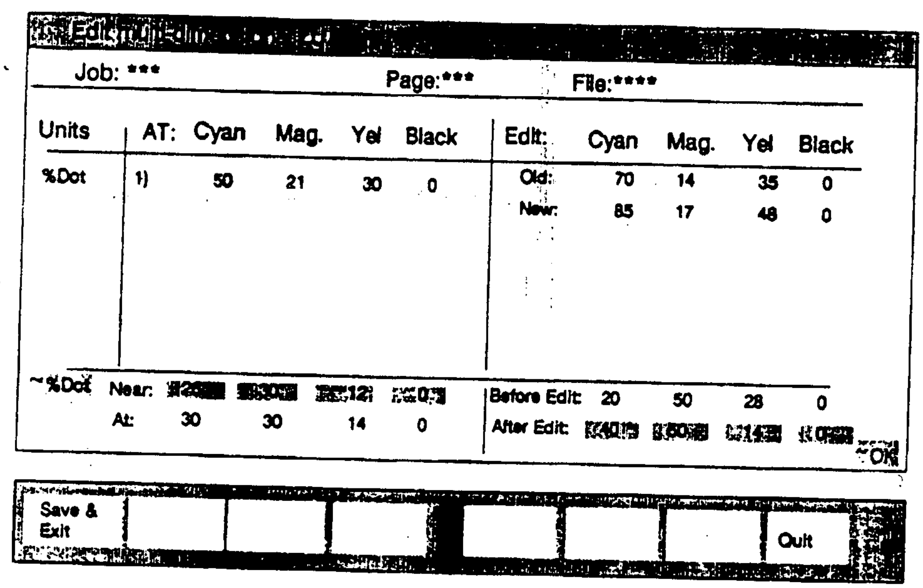
COORDINATES SCITEX TESTABLE

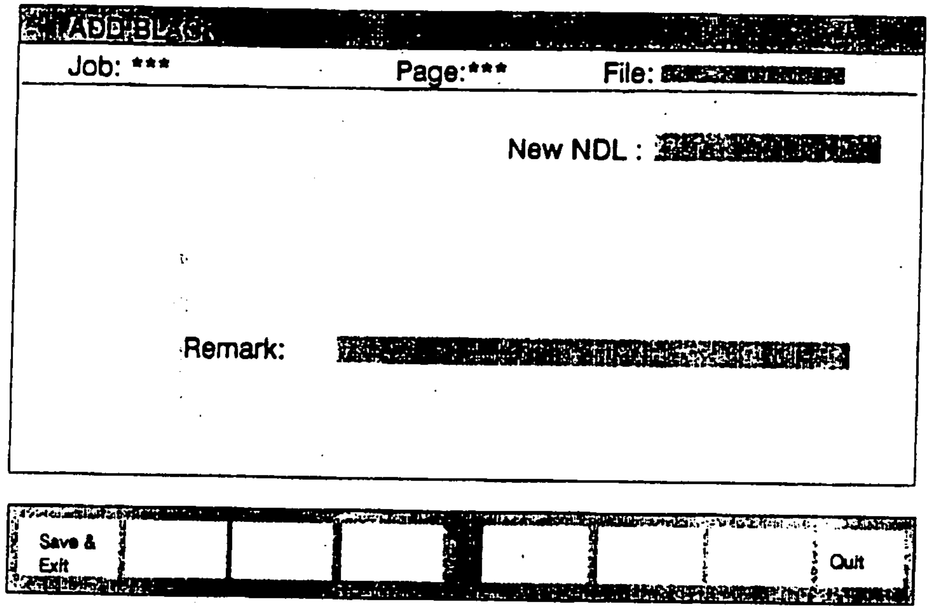



Job: ***
Page:***
File:
Colorimeter:
X-RITE 918
Receive Measures
Quit

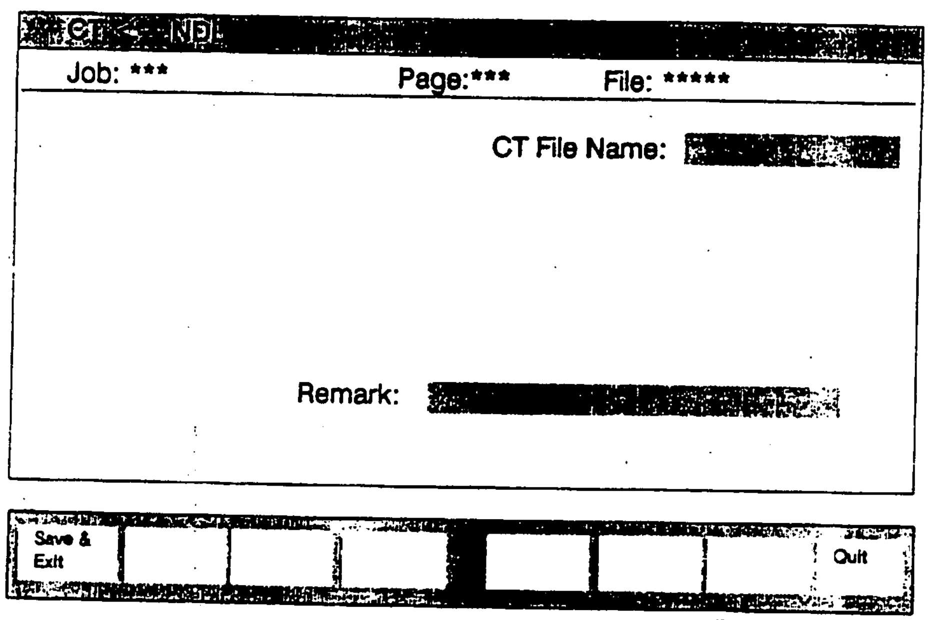

Job: ***
Page:***
File:
Proofer Measurements File Name:
Color Transform File Name:
Printer Measurements File Name:
Remark:
Save & Exit
Analyze
Edit
Quit


D LAB 0.........1.........2.........3.........4.........5.........6.........7.........8.........9.........10 MAX
0 - 1:
1 - 2:
2 - 3:
3 - 4:
5 - 6:
6 - 7:
7 - 8:
8 - 9:
9 - 10 :
10 - 12:
12 - 13 :
13 - 14:
14 - 15 :
15 - 16:
16 - ...:
Score is 8 , average is 2.1 and worst is 8.9 at (100,100,29,0)
Quit

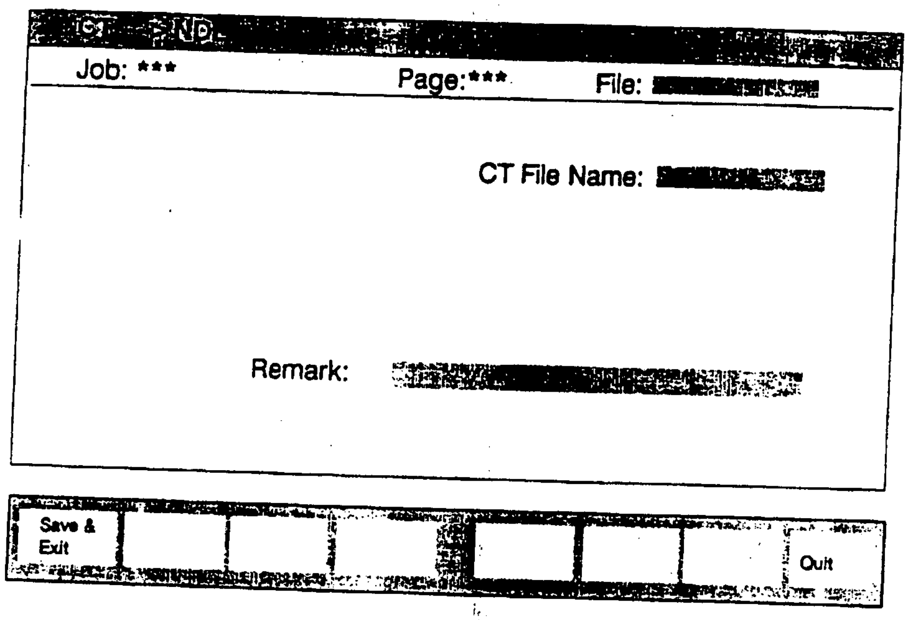

Job: ***
Page:***
File:****
Units
AT: Cyan Mag. Yel Black
%Dot
1) 50 21 30 0
Edit: Cyan Mag. Yel Black
Old: 70 14 35 0
New: 85 17 48 0
~%Dot
Near:
At: 30 30 14 0
Before Edit: 20 50 28 0
After Edit:
OK
Save & Exit
Quit

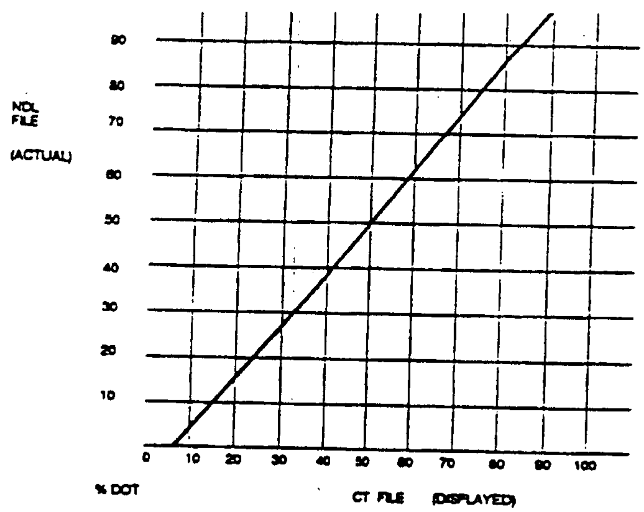


Job: ***
Page:***
File:
New NDL :
Remark:
Save & Exit
Quit



Job: ***
Page:***
File: *****
CT File Name:
Remark:
Save & Exit
Quit


Job: ***
Page:***
File:
CT File Name:
Remark:
Save & Exit
Quit

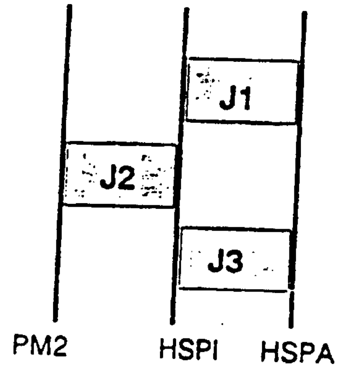

NDL FILE (ACTUAL)
90
80
70
60
50
40
30
20
10
0
10 20 30 40 50 60 70 80 90 100
% DOT
CT FILE (DISPLAYED)

Items	Quantity	Catalog No.
HSP-A	1	503D25747
HSP-I	1	503D26308
PM2 (4 Mb)	1	503D27365
Jumper	3	503B25748
Manual	1	399Z74725

Board	Location	Switch Setting
HSP-I	U102	1-7: ON; 8:OFF
PM2	U1105	1:ON; 2-4: OFF

	I/O, KEYLINE, ASSEMBLER, RIGHTOUCH, STARTER, SOFT/DTS	SOFT/DRUM	UNIFIED SYSTEMS ALL CONFIGURATIONS
PM2	SLOT 2	SLOT 15	SLOT 6
HSPI	SLOT 3	SLOT 16	SLOT 5
HSPA	SLOT 4	SLOT 17	SLOT 4

Table 1. HSP Slot Assignments for Whisper based Systems



HSP	MICRO-ASSEMBLER	VIEWER	MICRO-WHISPER
PM2	SLOT 9	SLOT 8	SLOT 5
HSPI	SLOT 8	SLOT 7	SLOT 6
HSPA	SLOT 7	SLOT 6	SLOT 7

Table 2. HSP Slot Assignments for Micro-Response Systems

Board Configuration

Functionality Enhancement

PM1 ~No

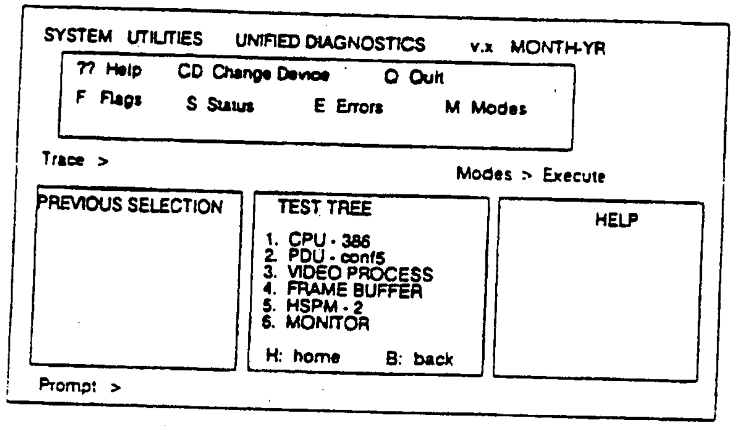
PM2 ~Yes

HSP ~Yes

Save & Exit | Enter | Main Menu | Quit

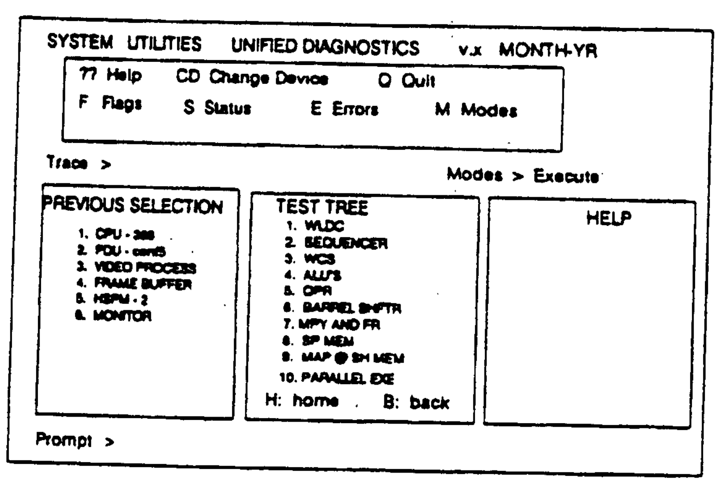
```
              SCITEX - DIAGNOSTICS SYSTEM  (CPU X86)

                  VERSION X.X  MONTH  YEAR

Opt. Available:   TT - Tutorial    SC - Set config    VH - Version Highlight
                  UT - Utilities    Q - Quit

Dev. Available:   W - WHISPER      U - UNIFIED      S - SMART_SCANNER
                    X - CSS_I      C - CSS_II        R - RAYSTAR
                    D - DOLEV      I - INTERFACE

                              Your Request:
```

HSP Installation v



vi

HSP Installation



APPENDIX F

Dump of file [YOAV_B.COLOR.COMPUTER.NDAT]PAT1.OBJ;8
on 31-OCT-1990 14:53:14.78 File ID (6574,5,0)
End of file block 24 / Allocated 24

Virtual block number 1 (00000001), 512 (0200) bytes

3A34 3120 3039 3931 2D54 434F 2D31 3330
2E31 5604 3154 4150 0402 0000 0000 0031
3356 2043 2058 4156 0100 0010 0000 0000
0000 0000 0000 0000 0000 0000 0000 3732
5009 6001 0500 0900 63FC 5014 0104 0000
43C8 FC50 0001 0402 01FA 3135 302D 312E
046C 6A05 AF37 BD36 866C 6A05 AF37 BD36
866C 6A05 AF37 BDB6 86E8 5000 0004 20FF
09A0 0105 666F 6F72 70FB 5006 0504 1B00
0504 5009 9C01 0574 6E69 7270 FB50 0005
0105 657A 6973 FC50 0F05 041B 0C05 0450
09A4 0105 7878 FE50 0C05 041B 0605 0450
7266 FC50 1905 041B 1405 0450 09AC 0105
7469 6E69 FC50 1405 041B 0F05 0450 09A8
5022 0504 1B1E 0504 5009 B401 0574 6573
FD50 1E05 041B 1905 0450 09B0 0105 6565
0504 1B26 0504 5009 BC01 0564 6E69 FD50
2605 041B 2205 0450 09B8 0105 6C6C 61FD
1B2E 0504 5009 C401 056D 6964 FD50 2E05
041B 2A05 0450 09C0 0105 6675 62FD 502A
0504 5009 CC01 0562 6772 FD50 3605 041B
3205 0450 09C8 0105 7475 6FFD 5032 0504
5009 D401 057A 7978 FD50 3E05 041B 3A05
0450 09D0 0105 796D 63FD 503A 0504 1B36
7475 6FF0 5046 0504 1B45 0504 1B42 0504
5009 D801 0578 78FE 5042 0504 1B3E 0504
6564 6E69 2079 6D63 2026 2062 6772 F150
5705 046E 6F69 736E 656D 6964 2074 7570
0472 6566 6675 6220 796D 63F6 5072 0504
7265 6666 7562 2062 6772 F650 6705 0478
6675 6220 7A79 78F6 5000 8D05 0578 6564
6E69 2079 6D63 2026 2062 6772 F150 7D05

```
Virtual block number 2 (00000002), 512 (0200) bytes

4320 7475 7074 754F F050 00A2 0505 7865
646E 6920 7A79 78F7 5000 9805 0572 6566
4080 C49C 4000 20C5 4100 20C5 4080 0000
4220 EC50 0A84 0105 656C 6261 5420 594D
2072 6566 6F6F 7270 2074 6573 2064 6E61
2074 696E 69D9 5000 B305 0502 0200 8312
7461 642E 6970 7766 F750 00DB 0505 2979
6D63 2064 6E61 2062 6772 202C 7A79 7828
6566 6675 6220 6267 72F6 5000 F005 0572
6566 6675 6220 7A79 78F6 5000 E505 0561
7865 646E 6920 6267 72F7 5001 0605 0572
6566 6675 6220 796D 63F6 5000 FB05 0572
207A 7978 2820 7265 746E 6972 7020 7465
7320 646E 6120 7469 6E69 E150 0110 0505
7562 207A 7978 F650 013A 0505 6274 6164
2E69 7077 66F7 5001 3005 0529 796C 6E6F
F6AD 3260 00D0 500C BCDE F8AD 01D0 F6AD
00B0 5E0C C200 0080 5018 0004 7265 6666
CF02 FB04 ACDD 0161 40DD 0151 08BC DE50
F6AD 325E 1360 9550 6041 D050 08BC DE51
50F8 AD01 7861 50D0 50F8 AD61 C151 0CAC
D00C 1201 F4AD B1F4 AD50 B050 50F7 0093
DE51 F6AD 32F6 AD50 B050 01F6 ADA1 0450
FFFF FFFF 8FD0 0812 01F8 ADD1 F8AD 50D0
04AC 9129 132C 04AC 915E 04C2 0000 0404
5000 8F98 A212 AE60 9550 6041 D050 08BC
1309 04AC 910B 133F 04AC 9111 132F 04AC
9117 133D 04AC 911D 1329 04AC 9123 1328
1C00 0104 9E5E 10C2 0004 0404 5050 D050
0000 0004 EC98 04AC 2090 0150 D507 1105
0450 01D0 0412 2AF8 BD91 FFF4 CD08 ACD0
F2AD FFFF 8FB0 FFFF FFF8 ED04 ACD0 529D
```

Virtual block number 3 (00000003), 512 (0200) bytes

```
DD01 50F8 BD98 0450 50D0 50F2 AD32 0812
F8BD 9504 5050 D050 F2AD 3208 12F4 BD95
0960 C2F8 BD89 0150 D500 8B31 F8AD 50D0
5001 F8AD C10F 1220 5091 FF6C CF01 FB50
D51B 1105 1325 F4BD 910B 1325 F8BD 9111
1350 5191 5009 60C2 8002 0200 F4BD 8951
D050 01F8 ADC1 4811 F2AD 01B0 F4AD 50D0
5001 F4AD C1F8 AD50 D050 01F8 ADC1 0150
D050 01F8 ADC1 0125 13F8 BD95 2A13 2050
91FF 07CF 01FB 50DD 0150 F8BD 98F8 AD50
0611 0150 D5DF 1105 13F8 BD95 0A13 2050
91FE E7CF 01FB 50DD 0150 F8BD 98F8 AD50
6056 5000 B0DE 5004 BCDE 5112 010C ACD1
5E20 C201 FC04 FF80 2531 01FF 3531 50D5
FFFF FFFF 8FD0 0D18 FE07 CF50 7150 6056
5000 B0DE 5004 BCDE 1F15 FE11 CF50 7150
5250 6650 6056 5000 B0DE 5004 BCDE 5260
5650 08BC DE54 10BC DE01 50D5 1F11 0450
52EA AD32 EAAD 00B0 E8AD E4AD B0E4 AD00
B0F8 AD00 0000 008F 5004 5000 D064 5270
F4AD 56F4 AD62 4450 5262 43D0 5204 BCDE
53EA AD32 54E8 80AD 3201 4A1E 0CAC 52D1
ADA1 E6AD EAAD B0F8 ADF4 AD50 0A15 F8AD
F4AD 51F4 AD52 5052 F4AD 5208 1852 7352
FFFF 8FD0 0818 FD6E CF52 7152 F8AD 56B7
1F0C AC52 D152 EAAD 32EA AD52 B052 01EA
53E6 AD32 6462 43D0 5204 BCDE 53E6 AD32
5404 BCDE F0AD 62D0 5204 BCDE 0450 FFFF
4350 5208 BCDE 53E6 AD32 5408 BCDE ECAD
6250 5208 BC85 DE62 43F0 ADD0 5204 BCDE
00BB 3103 1F0C AC52 D152 EAAD 32EA AD01
B062 43EC AD50 5208 BCDE 53E6 AD32 6462
```

Virtual block number 4 (00000004), 512 (0200) bytes

```
5266 5262 5652 00B2 DE52 04BC DE50 5256
5262 5252 6243 D052 04BC DE53 EAAD 3201
EAAD 3254 E8AD 3201 50D5 4B1E 0CAC 52D1
52E8 AD32 E8AD 01B0 F4AD 5250 5250 7650
00B2 DE52 04BC DE53 E8AD 3255 F4AD 5657
6044 5650 6342 8002 0200 D053 04BC DE52
D152 E8AD 32E8 AD52 B052 01E8 ADA1 6044
5250 5257 7657 5560 5552 6452 6243 5652
6056 5264 5262 5652 08BC DE56 F4AD 5650
6544 5655 08AC D054 EAAD 32B8 1F0C AC52
B0FF 4631 031E 0CAC 52D1 52EA AD32 EAAD
52B0 5201 EAAD A165 4452 5052 5076 5056
A164 4352 D052 0464 43C1 5404 ACD0 53EA
AD32 241E 0C80 AC52 D152 EAAD 32EA AD01
0152 0CAC 01C3 52DD 0152 0810 ACC1 DC1F
0CAC 52D1 52EA AD32 EAAD 52B0 5201 EAAD
ADB5 E2AD 52B0 5250 F7FD BBCF 04FB 52DD
0152 0404 ACC1 52DD 0152 0408 ACC1 52DD
EAAD 32EA AD01 B064 5270 5262 5652 08BC
DE54 10BC DE04 5052 D052 E2AD 3208 13E2
3254 6243 5652 00B2 DE52 04BC DE53 EAAD
3256 10AC 80D0 0150 D53D 1E0C AC52 D152
D152 EAAD 32EA AD52 B052 01EA ADA1 6652
7052 6654 6354 6243 6452 10BC DE53 EAAD
0000 8F32 00A4 5270 5264 5267 5262 5652
00B2 DE52 04BC DE54 10AC D0C6 1F0C AC52
D008 1B0A 10AC D104 50FF FFFF FF8F D008
1B10 AC0C ACD1 5EFE 18CE 9E00 0404 0450
0150 D536 1E50 51D1 5010 AC10 ACC5 51FE
1ACD 3285 FE1A CD00 B004 50FF FFFF FF8F
52B0 5201 FE1A CDA1 FE44 CD42 5050 5060
4176 5004 BCDE 51FE 1ACD 3252 FE1A CD32
```

Virtual block number 5 (00000005), 512 (0200) bytes

```
32FE 1ACD FE18 CDB0 FE18 CD00 B0CD 1F52
50D1 5210 AC10 ACC5 50FE 1ACD 32FE 1ACD
C052 0452 C552 FE18 CD32 50FE 44CD DE51
FE1A CD32 50D5 601E 0CAC 52D1 52FE 1ACD
4152 5052 6240 7652 08BC DE50 FE1A CD32
51FE 1ACD 9A02 0200 32D4 AD41 50D0 5052
52B0 5252 F752 10AC 52C1 5252 D052 FE18
CD32 FE1A CD52 B052 01FE 1ACD A1FE 1CCD
FBD4 ADDF 01FE 1CCD DF01 0CAC DD01 14AC
DD01 A21F 0CAC 52D1 52FE 1ACD 32FE 18CD
C2DF 0109 98C2 DF01 5280 1C00 0104 9E5E
10C2 000C 0150 D504 0450 50D0 FC34 CF04
011B 11F7 AD00 9007 1350 D550 0998 C2FF
FFFF FE8F CBFA 51CF 03FB 04AC DD01 099C
FFFF FEFF 8FCB F7AD 0994 C290 0611 F7AD
0190 0613 50D5 5009 98C2 FFFF FFFD 8FCB
C240 9E50 06C4 50F7 AD9A 1113 50D5 5009
98C2 FFFF EFFF 8FCB 6013 50D5 5009 98C2
C240 9E50 06C4 50F7 80AD 9A11 1350 D550
0998 C2FF FFDF FF8F CB60 08AC B050 098A
88C2 409E 5006 C450 F7AD 9A11 1350 D550
0998 C2FF FFBF FF8F CB60 08AC B050 098A
409E 5006 C450 F7AD 9A16 1350 D550 0998
C2FF FFFB FF8F CB01 3531 6008 ACB0 5009
8FCB 0100 3103 1250 D550 0998 C2FF FFFD
FF8F CB50 D501 1331 6008 ACB0 5009 8CC2
0CEF 4A1E 0350 D180 50F6 AD9A F8AD 01D0
F6AD 0090 6113 50D5 5009 98C2 FFFF FFF7
9A50 03C4 50F7 AD9A 1513 50D5 50F8 AD50
CB50 50D2 50FF FFFF F08F CA50 0998 C214
9AF8 AD50 D050 F8AD 0178 F6AD 5090 5001
F6AD 8109 62C2 4108 ACB0 5150 C051 F6AD
```

Virtual block number 6 (00000006), 512 (0200) bytes

```
51F7 AD9A 1C13 50D5 5009 98C2 FFFF EFFF
8FCB 0150 D500 9431 B61F 0350 D150 F6AD
C2FF FFDF FF8F F0CB 0450 50D0 5009 70C2
41C0 5002 50C5 5003 C450 08AC 3C51 03C4
C240 9E50 06C4 50F7 AD9A 5108 AC3C 5303
C453 9D02 0200 F7AD 9A2F 1350 D550 0998
FFFF BFFF 8FCB 0450 50D0 5009 74C2 43C0
5002 51C5 5150 C450 603C 5060 3E50 098C
5009 78C2 41C0 5002 50C5 5003 C450 08AC
3C51 03C4 51F7 AD9A 1C13 50D5 5009 98C2
9CC2 DF01 0998 C2DF 0152 801C 0001 049E
5E0C C200 0404 0450 0000 8F32 0450 50D0
CB01 1B11 F9AD 0090 0713 50D5 5009 98C2
FFFF FFFE 8FCB F859 CF03 FB04 ACDD 0109
C2FF FFFE FF8F CBF9 AD09 94C2 9006 11F9
AD01 9006 1350 D550 0998 C2FF FFFF FD8F
8AC2 409E 5006 C450 F9AD 9A17 1350 D550
0998 C2FF FFEF FF8F CB6F 1350 D550 0998
C450 F9AD 9A17 1350 D580 5009 98C2 FFFF
DFFF 8FCB 0450 50D0 5060 3C50 603E 5009
1713 50D5 5009 98C2 FFFF BFFF 8FCB 0450
50D0 5060 3C50 603E 5009 8AC2 409E 5006
0998 C2FF FFFB FF8F CB04 5050 D050 603C
5060 3E50 0988 C240 9E50 06C4 50F9 AD9A
FFF7 8FCB 0450 50D0 5060 3C50 603E 5009
8CC2 409E 5006 C450 F9AD 9A17 1350 D550
F9AD 9A0E 1350 D550 A509 98C2 FFFF EFFF
8FCB FAAD 00B0 6013 50D5 5009 98C2 FFFF
5003 C450 F9AD 9A0E 1350 D550 0998 C2FF
FFDF FF8F CBFA AD09 62C2 40B0 5003 C450
66C2 40B0 5003 C450 F9AD 9A0E 1350 D550
0998 C2FF FFBF FF8F CBFA AD09 64C2 40B0
```

Virtual block number 7 (00000007), 512 (0200) bytes

```
ADFF FFFF FF8F D053 CC1C 0001 049E 5E1C
C200 0C01 0404 5050 D050 FAAD 3CFA AD09
52D5 5209 98C3 FFFF FFBF 8FCB F6F1 CF03
FB04 ACDD 0109 9CC3 DF01 0998 C3DF 01F0
FD8F CB50 D522 1109 94C3 0090 0913 52D5
8002 0200 5209 98C3 FFFF FFFE 8FCB 3B13
AD00 D004 50FF FFFF FF8F D001 50D5 0B11
0994 C301 900A 1352 D552 0998 C3FF FFFF
D252 0998 C3FF FFFF F08F CB01 A131 031F
0252 D152 EEAD 9AF4 AD01 D0EE AD00 90F0
AD00 9001 015C 3104 1252 D552 0998 C3FF
FFFF 7F8F CB12 1252 D552 F4AD 52CB 5252
50FF FFFF F08F CA50 0998 C314 0CEF 01B4
4731 031F 0352 D152 EFAD 9AF8 AD01 D0EF
8FCB 0100 FE31 0412 50D5 5009 98C3 FFFF
FF7F 8FCB 1212 50D5 50F8 AD50 CB50 50D2
1100 0502 0050 CF50 51D0 5150 D050 EFAD
9A00 BF31 0312 50D5 5009 98C3 FFFF FFEF
B050 0988 C340 9E50 06C4 50EE AD9A 50D5
2A11 50D5 3FFF F5E6 EF17 0109 1180 500A
03C4 51EE AD9A 50D5 0211 6000 B050 098A
C340 9E50 06C4 50EE AD9A 50D5 1611 6000
FF8F B051 50C0 51EF AD9A 5003 C450 EEAD
9A16 1209 62C3 40B5 5051 C050 EFAD 9A51
51C0 50EF AD9A 5103 C451 EEAD 9A52 50C0
52EF AD9A 5003 C450 EEAD 9A09 62C3 41FF
C342 D509 70C3 4250 D0BF 1C43 4F4C 4C41
4D06 0001 FB50 DD01 5009 62C3 F740 3C50
50D5 5009 98C3 FFFF FFDF 8FCB 50D5 2F11
F0AD 00D0 0450 FFFF FFFF 8FD0 0812 0970
B21C 4545 5246 0400 01FB 0970 C341 DD01
5150 C051 EFAD 9A50 03C4 50EE AD9A 1F13
```

Virtual block number 8 (00000008), 512 (0200) bytes

```
3103 1E03 50D1 50EF AD9A F8AD 50D0 50F8
AD01 78EF AD50 9050 01EF AD81 F0AD 00D0
FE5F 3103 1E02 52D1 52EE AD9A F4AD 52D0
52F4 AD01 78EE AD52 9052 01EE AD81 FEB9
DD01 FB1C 3154 5845 0400 01FB 10AC FC02
0200 DD01 5E10 C200 0004 0450 FFF0 CDD0
5050 D050 6140 3C51 14BC DE50 F8AD D0FF
F8CD 008F 98E2 1C32 5458 4504 0001 FB04
5458 4504 0001 FB11 DD01 DEF8 AD03 F3F6
1C33 5458 4504 0002 FBF8 ADDD 0150 DD01
50D0 5008 A041 3C50 14AC D051 F4AD 50C1
50F8 AD04 C5F4 AD00 D0F8 AD00 D0D5 1C34
CAF8 AD10 F3D3 F4AD 03F3 F01C 3554 5845
0400 03FB F8AD DD01 F4AD DD01 50DD 0150
F91C 0005 049E 5E38 C200 3C04 0450 0000
8F32 F31C 3654 5845 0400 01FB 04AC DD01
ADDD 0154 04BC DE53 F4AD D0FF F4CD 00D0
ED1C 3731 5458 4505 0001 FB08 ACDD 0155
F8AD 00D0 FFF4 CD00 D0E1 F4AD 03F3 6443
52B0 5250 F7D3 1C39 5458 4504 0001 FBF4
1C30 3154 5845 0500 02FB F4AD DD01 F8AD
DD01 5204 ACD0 53F8 AD50 C150 F4AD 04C5
5250 F7B3 1C38 5458 4504 0000 FBCD F4AD
10F3 D6F8 AD03 F308 A243 50B0 5050 F7EC
52DC AD3C DCAD 50B0 FB47 CF01 FB57 A5DF
01F9 58CF 02FB 46A5 DF01 52DD 0152 523C
F927 CF02 FB72 A5DF 0152 DD01 52DC AD3C
ECAD 50D0 F93B CF02 FB67 A5DF 0152 DD01
B001 50D5 00D1 3103 1950 F4AD D1E9 1C32
3154 5845 0500 00FB F4AD 00D0 E8AD 50D0
FB52 DEAD 3C50 D547 1850 E0AD D1F2 1C31
3154 5845 0500 00FB E0AD 00D0 DEAD F4AD
```

Virtual block number 9 (00000009), 512 (0200) bytes

```
FB50 DD01 50DE AD3C E0AD DD01 52E0 ADC0
5250 C450 50D0 E91C 3131 5458 4505 0000
4505 0000 FBE0 AD50 D050 01E0 ADC1 E8BD
4250 B050 50F7 ED1C 3331 5458 4505 0002
ADD1 F21C 3854 5845 0400 00FB FE02 0200
E0AD 00D0 BB19 50E0 ADD1 F61C 3131 5458
DD01 52E0 ADC0 5250 C450 50D0 E91C 3854
5845 0400 00FB 52DE AD3C 50D5 4718 50E0
01E0 ADC1 ECBD 4250 B050 50F7 ED1C 3431
5458 4505 0002 FB50 DD01 50DE AD3C E0AD
00FB F4AD 52D0 5201 F4AD C1BB 1950 E0AD
D1EF 1C38 5458 4504 0000 FBE0 AD50 D050
53DD 0153 52C0 52DC AD3C 53DE AD3C FF32
3103 1850 F4AD D1C8 1C32 3154 5845 0500
FBF0 AD00 D0E4 AD50 D0F8 19CF 02FB 008D
C5DF 01F0 ADDD 01F8 27CF 02FB 7DA5 DF01
D1E9 1C31 3154 5845 0500 00FB F8AD 00D0
5318 50F0 ADD1 F41C 3531 5458 4505 0000
C1F5 1C36 3154 5845 0500 03FB E4AD DD01
F0AD DD01 F8AD DD01 0150 D52C 1850 F8AD
52D0 5201 F0AD C1D7 1950 F8AD D1EF 1C31
3154 5845 0500 00FB F8AD 50D0 5001 F8AD
A3CF 02FB 0098 C5DF 01F0 ADDD 01AD 1950
F0AD D1DF 1C35 3154 5845 0500 00FB F0AD
AD02 C551 04AC 18C5 F8AD 00D0 5380 1C00
0104 9E5E 08C2 000C 0404 5000 008F 32F7
C562 60B0 5020 A340 9E50 51C0 50F8 AD02
C551 08AC 18C5 5220 A340 9E50 51C0 50F8
603E 5018 A340 9E50 51C0 50F8 AD02 C551
08AC 18C5 5250 C052 F8AD 02C5 5004 AC18
A340 9E50 08AC 18C5 5128 A340 9E50 04AC
18C5 AAF8 AD03 F360 61B0 5018 A342 9E51
```

Virtual block number 10 (0000000A), 512 (0200) bytes

```
6260 B052 2CA3 419E 5062 3E52 D12C A340
9E50 08AC 18C5 5104 AC18 C561 6250 5228
C200 0C04 0000 C260 B052 2EA3 419E 5062
3E52 2EA3 409E 5008 AC18 C551 04AC 18C5
3250 20A2 409E 5051 C050 BB02 0200 0CAC
02C5 5108 AC18 C552 F61C 0001 049E 5E08
ADD5 F8AD 53D0 5350 C250 6032 5020 A240
9E50 51C0 500C AC02 C551 04AC 18C5 5360
801C 0001 049E 5E18 C200 0404 0450 50D0
50FF F8CD 3CF8 AD50 D050 F8AD CE08 18F8
AD18 C501 50D5 0153 3103 1F5C F8AD D15C
14A2 3CF8 AD00 D009 E0C2 5CB0 5C62 4952
14A2 3CF4 AD5C D05C 01F8 ADC1 0123 3103
1A5C D55C 6C3C 5C6C 3E5C 2EA2 4C9E 5CF8
1A5C D55C 6C3C 5C6C 3E5C 2EA2 4C9E 5CF4
AD18 C501 50D5 010D 3103 1F5C F4AD D15C
186C 6051 5C28 A24C 9E5C F8AD 18C5 5028
A24C 9E5C F4AD 18C5 F0AD 00D0 00DD 3103
3CFE 4FCF 02FB F8AD DD01 F4AD DD01 FE5C
CF02 FB5C DD01 5C14 A23C F8AD DD01 2D80
DD01 ECAD DD01 481E 04A2 ECAD D1EC AD00
D0FE 3FCF 02FB F4AD DD01 5CDD 015C 14A2
F0AD D150 D45C 09E0 C23C F0AD 5CD0 5CF0
ADC0 5C50 3CFE D7CF 03FB F8AD DD01 F4AD
04A2 ECAD D1EC AD5C D05C 01EC ADC1 50D5
1211 0413 E8AD D5E8 AD50 D050 D602 1B5C
1A6C 60B1 5C2C A24C 9E5C F4AD 18C5 502C
A24C 9E5C F8AD 18C5 3B12 E8AD D5B8 891F
6C00 B05C 2EA2 4C9E 5CF8 AD18 C501 50D5
1311 6C00 B05C 2EA2 4C9E 5CF4 AD18 C512
5CD0 5C01 F8AD C1FE F631 031E 5CF4 ADD1
5C14 A23C F4AD 5CD0 5C01 F4AD C101 1711
```

Virtual block number 11 (0000000B), 512 (0200) bytes

```
9E51 FE1C 0006 049E 5E08 C200 0001 50D5
04FE B031 031E 5CF8 ADD1 5C14 A23C F8AD
22A1 4C9E 5CF8 AD24 C501 50D5 231E 5CF8
ADD1 5C16 A03C F8AD 00D0 50D1 1C00 0104
0401 50D5 04E0 1F5C EA02 0200 F8AD D15C
16A0 3CF8 AD5C D05C 01F8 ADC1 6C00 B05C
D008 1262 D562 50D0 E61C 434F 4C4C 414D
0600 01FB 08AC DD01 5204 ACD0 5E04 C200
B052 921C 0001 049E 53FE 1C00 0504 9E5E
08C2 000C 0404 5000 D004 50FF FFFF FF8F
50A4 5009 E2C2 403C 50FA AD3C 51F8 AD3C
271E 0450 D150 FAAD 3CFA AD00 B0F8 AD01
50F8 AD3C 00A2 C3DF 01D9 1F04 50D1 50FA
AD3C FAAD 50B0 5001 FAAD A1F8 AD51 B051
D004 50FF FFFF FF8F D008 1350 D5FF 71CF
03FB 04AC DD01 50DD 0150 0CA2 C450 02C4
00B0 FEFE CD00 B053 801C 0001 049E 55FE
1C00 0604 9E5E FEF8 CE9E 003C 0404 5000
FEFC CD3C 0150 D554 1E16 A3FE FCCD B1FE
FCCD 00B0 FF00 CD96 7F4C 188F 50FE F8CD
18FF 00CD 6251 521E A540 9E50 24C4 50FE
FCCD 3C2D 1362 B552 22A5 409E 5024 C450
5201 FEFC CDA1 FF00 CD62 5052 1EA5 409E
5024 C450 FEFE CD3C FEFE CDFE FCCD B019
50D5 411E 10A3 5280 D152 FEFC CD3C FEFC
CD00 B0AF 1F16 A3FE FCCD B1FE FCCD 52B0
18C4 5262 3C52 623E 5265 429E 5250 C052
02C4 52FE FCCD 3C50 24C4 50FE FECD 3C01
B0C2 1F10 A352 D152 FEFC CD3C FEFC CD52
B052 01FE FCCD A162 02B0 522E A342 9E52
14A3 FEFA CDB1 FEFA CD00 B001 50D5 008B
3103 1F08 A352 D152 FEFC CD3C FEFC CD00
```

Virtual block number 12 (0000000C), 512 (0200) bytes

```
CDB0 341B 0150 C4D1 5060 3C50 603E 502E
A340 9E50 18C4 50FE FACD 3C01 50D5 621E
02C4 50FE FCCD 3C51 18C4 51FE FACD 3C52
513C FEF8 CD50 B050 01FE F8CD A151 FEF8
F152 0000 01FF 0305 01FD FEFA CDA1 FF08
CD42 5070 5060 6D50 20A3 409E 5051 C050
3232 5F6C 6163 6F6C 08ED 1B00 0000 1800
0600 BE0F 3154 4150 0400 0000 0700 BC0B
0044 00BF 0633 325F 6C61 636F 6C08 ED1B
0000 00A0 0006 00BE 0F00 0000 8800 BF06
00BE 1000 0000 F000 BF06 3432 5F6C 6163
6F6C 08ED 1B00 0000 E400 0600 BE0F 0000
04A4 0006 00BE 1000 0002 D000 BF06 325F
325F 6C61 636F 6C09 EC1B 0000 01D4 0006
EC1B 0000 05A8 0006 00BE 1000 0001 0100
BF06 315F 325F 6C61 636F 6C09 EC1B 0000
636F 6C09 EC1B 0000 07A0 0006 00BE 1000
0001 F800 BF06 315F 315F 6C61 636F 6C09
315F 6C61 636F 6C09 EC1B 0000 0900 0006
00BE 1000 0001 5F00 BF06 325F 315F 6C61
BF06 365F 335F 6C61 636F 6C09 EC1B 0000
0B2C 0006 00BE 1000 0002 2C00 BF06 345F
0002 4800 BF06 355F 335F 6C61 636F 6C09
EC1B 0000 0BC4 0006 00BE 1000 0000 9800
0006 00BE 0E00 0000 B400 BF06 315F 6C61
636F 6C07 EE1B 0000 0E0C 0006 00BE 0E00
EE1B 0000 0F14 0006 00BE 0E00 0000 5400
BF06 325F 6C61 636F 6C07 EE1B 0000 0EC0
6C61 636F 6C08 ED1B 0000 1090 0006 00BE
0F00 0001 7900 BF06 345F 6C61 636F 6C07
00BF 0635 5F6C 6163 6F6C 07EE 1B00 0010
D800 0600 BE0E 0000 0045 00BF 0630 325F
```

Virtual block number 13 (0000000D), 512 (0200) bytes

BE0E	0000	0078	00BF	0636	5F6C	6163	6F6C
07EE	1B00	0011	0000	0600	BE0E	0000	0028
0013	AC00	0600	BE0E	0000	0233	00BF	0637
5F6C	6163	6F6C	07EE	1B00	0011	7800	0600
FACD	50B0	5001	8002	0200	0000	BE0E	0000
00E4	00BF	0638	5F6C	6163	6F6C	07EE	1B00
3103	1E08	A352	D152	FEFC	CD3C	FEFC	CD52
B052	01FE	FCCD	A1A1	1F14	A3FE	FACD	B1FE
CD52	B052	01FE	F8CD	A150	FEF8	CDB0	2F1E
10A3	52D1	52FE	FCCD	3CFE	FCCD	00B0	FF78
1F10	A352	D152	FEFC	CD3C	FEFC	CD52	B052
01FE	FCCD	A1FF	08CD	4208	7052	503C	FEF8
08BC	DE50	FE96	FCCD	3C54	FEFC	CD3C	012E
1E08	A352	D152	FEFC	CD3C	FEFC	CD00	B0D1
D31F	08A3	52D1	52FE	FCCD	3CFE	FCCD	52B0
5201	FEFC	CDA1	D0AD	4450	7050	6240	6D52
FBFF	08CD	DF01	D0AD	DF01	10A3	DD01	10A3
DD01	04AC	DD01	D0AD	4208	7052	FEFC	CD3C
7C01	0404	5000	008F	3204	50FF	FFFF	FF8F
D008	13FF	04CD	D5FF	04CD	50D0	F113	CF05
B001	50D5	00C1	3103	1F0C	A352	D152	F6AD
3CF6	AD00	B053	801C	0001	049E	5E10	C200
409E	5018	C450	F4AD	3C01	6D1E	14A3	F4AD
B1F4	AD00	B0F8	AD00	0000	008F	50F2	AD00
3CF2	AD50	B050	01F2	ADA1	51F2	ADB0	55F8
AD56	451B	0150	D150	603C	5060	3E50	2EA3
3C50	603E	5018	A340	9E50	51C0	5002	C450
F6AD	3C51	18C4	51F4	AD3C	5408	BCDE	5251
F4AD	B1F4	AD50	B050	01F4	ADA1	F8AD	5050
5055	7655	50A1	6050	6442	6450	506E	5060
0060	50F8	AD56	5404	BCDE	52F6	AD3C	F8AD
5050	50F8	AD08	4309	18F8	AD53	941F	14A3

Virtual block number 14 (0000000E), 512 (0200) bytes

```
8F98 FF42 3103 1E0C A350 D150 F6AD 3CF6
AD50 B050 01F6 ADA1 6442 50B0 5050 6950
016F 1E0C A3F8 ADD1 F8AD 00D0 F4AD 00D0
53D7 1C00 0104 9E5E 10C2 001C 0404 5000
50F8 AD50 9E02 0200 C150 04C4 5061 4098
510C BCDE 50F8 ADD0 5404 BCDE 52F8 ADD0
C250 6140 3C51 08BC DE50 F8AD D052 6140
3C51 04BC DE50 F8AD D064 4209 EAC3 40B0
50D0 50F0 ADF4 ADC1 010B 11F4 AD50 D050
F4AD F0AD C30D 18F0 ADD5 F0AD 52D0 5250
0104 9E5E 10C2 001C 0404 50F4 ADD0 921F
0CA3 F8AD D1F8 AD50 D050 01F8 ADC1 F4AD
50F8 ADD0 5304 BCDE 52F8 ADD0 0163 1E0C
A4F8 ADD1 F8AD 00D0 F4AD 00D0 5481 1C00
6140 3C51 08BC DE50 F8AD D052 6140 3C51
04BC DE50 F8AD D063 4261 40B0 510C BCDE
ADF4 ADC1 0150 D50D 11F4 AD50 D050 F4AD
F0AD C30F 18F0 ADD5 F0AD 52D0 5250 C250
10C2 007C 0404 50F4 ADD0 9E1F 0CA4 F8AD
D1F8 AD50 D050 01F8 ADC1 F4AD 50D0 50F0
0150 D53F 1E03 50D1 50F2 AD3C F2AD 00B0
F4AD 0000 0000 8F50 5480 1C00 0104 9E5E
5250 6450 0A88 C440 5650 F2AD 3C52 506E
5060 413C 5004 BCDE 51F2 AD3C 55F4 AD56
5204 ACD0 C41F 0350 D150 F2AD 3CF2 AD50
B050 01F2 ADA1 F4AD 5050 5055 7655 5260
F4AD 5655 F8AD 56F8 AD50 5050 5576 5552
6452 0A94 C456 5550 6E50 603C 5006 A23E
5616 1852 7352 F4AD 56F4 AD50 5050 5576
5552 6452 0A84 C456 B955 5262 5214 6652
3C50 5269 5200 6252 F4AD 5601 50D5 1411
0450 50D0 5050 3C50 5269 5200 6052 F4AD
```

Virtual block number 15 (0000000F), 512 (0200) bytes

```
AD00 B0EC AD00 0000 008F 5056 BA1C 0001
049E 5E1C C200 7C50 D504 0450 50D0 5050
6041 3C50 10BC DE51 E6AD 3C53 04BC DE52
E6AD 3C50 D530 1E04 A652 D152 E6AD 3CE6
D1D2 8002 0200 1F04 A650 D150 E6AD 3CE6
AD50 B050 01E6 ADA1 6342 50B0 5050 F750
E6AD 00B0 06A3 52B0 5252 F752 503C FECF
CF01 FB0C ACDD 0153 04AC D017 1B03 08A6
08BC DE51 E6AD 3C52 6041 3250 04BC DE51
E6AD 3C01 50D5 5D1E 08A6 52D1 52E6 AD3C
5654 ECAD 56E8 AD50 5050 5276 5250 6650
6656 5250 5650 524E 5250 C250 6041 3250
ADBB 3CE6 AD50 B050 01E6 ADA1 ECAD 5050
5054 7654 5260 5250 6450 E8AD 5652 E8AD
0050 808F F0AD 6111 18F0 AD73 F0AD 5270
5208 A66E 54EC AD56 A61F 08A6 50D1 50E6
001C 0404 50FF FFFF ECED 50EC AD52 5052
5476 54F0 AD66 F0AD 5270 5200 0000 0000
1819 520C AC71 5262 5652 28A4 429E 5218
C452 0A98 C43C 5480 1C00 0104 9E5E 08C2
5218 C452 0A98 C43C 0401 1B52 D552 623C
5262 3E52 2EA4 429E 5218 C452 0A98 C43C
AD00 B062 14AC B052 2CA4 429E 5218 C452
0A98 C43C 6352 5052 0CAC 7653 28A4 429E
5051 C050 02C4 50FA AD3C 5118 C451 0A98
C43C 0150 D539 1E08 A452 D152 FAAD 3CFA
D150 FAAD 3CFA AD50 B050 01FA ADA1 6260
41B0 5004 BCDE 8051 FAAD 3C52 20A4 409E
C453 FAAD 3C50 18C4 500A 98C4 3C36 1E0C
A452 D152 FAAD 3CFA AD00 B0CA 1F08 A450
3CFA AD50 B050 01FA ADA1 6062 41B0 5018
A443 9E52 08BC DE51 FAAD 3C53 50C0 5302
```

Virtual block number 16 (00000010), 512 (0200) bytes

```
FAAD B1FA AD00 B062 01B0 522E A442 9E52
18C4 520A 98C4 3CCA 1F0C A450 D150 FAAD
B009 1250 00D1 5060 3C50 603E 502E A440
9E50 18C4 50CF FAAD 3C01 50D5 571E 14A4
D402 0200 5018 C450 FAAD 3C52 28A4 409E
5018 C450 0A98 C43C 0135 110A 98C4 FAAD
AC1F 14A4 FAAD B1FA AD50 B050 01FA ADA1
0A98 C4FA ADB0 0618 6062 5150 28A4 409E
031F 10A4 04AC D154 801C 0001 049E 56FE
1C00 0604 9E5E FEB8 CE9E 007C 50D5 0404
5201 C052 6243 3C52 0CBC DE53 04AC 01C3
1813 04AC D550 0CBC DE55 04AC D001 0531
3C53 6342 3C53 0CBC DE52 04AC D060 45FE
C2CD B0FE C2CD 00F7 0106 11FE C2CD 52F7
5060 423C 500C BCDE 5204 ACD0 0100 B131
031B 5253 D152 04AC C052 10A4 C252 14A4
500C BCDE 5204 AC80 D0FE DCCD 5050 5050
7650 10AC 6050 6256 5228 A440 9E50 18C4
1862 FEDC CD51 5218 BCDE 381B 50D5 5060
3C50 623E 522E A440 9E50 18C4 5060 423C
CF06 FB52 DD01 5201 04AC C108 ACDD 010C
ACDD 017E 5070 50FE DCCD 5618 ACDD 012D
0163 42A1 FEB8 CD04 ACD0 5204 ACD0 530C
ACD0 0450 FFFF FFFF 8FD0 0813 50D5 FF23
53D1 5204 ACC0 8052 10A4 C252 14A4 3C53
6042 3C50 0CBC DE52 04AC D063 4250 B050
D57B 1E08 A452 D152 FEC0 CD3C FEC0 CD00
B0FE BCCD 00B0 0103 4431 FF50 31[illegible]3 1A52
CD50 B050 01FE BCCD A151 FEBC C[illegible]0 521E
10A4 50D1 50FE BECD 3CFE BECD 0[illegible] 0150
5051 C050 02C4 50FE C0CD 3C51 1[illegible]4 5161
403C 510C BCDE 50FE BECD 3C52 51[illegible] FEBC
```

Virtual block number 17 (00000011), 512 (0200) bytes

```
D150 FEBE CDBD 3CFE BECD 50B0 5001 FEBE
CDA1 FEE0 CD42 5070 5060 6D50 20A4 409E
FEBE CD00 B088 1F08 A450 D150 FEC0 CD3C
FEC0 CD50 B050 01FE C0CD A1AE 1F10 A450
0200 5252 3CFE BCCD 50B0 5001 FEBC CDA1
52FE BCCD B02F 1E10 A452 D152 FEBE CD3C
00B0 D11F 10A4 50D1 50FE BECD 3CFE BECD
50B0 5001 FEBE CDA1 FEE0 CD42 0870 8002
6240 6D52 08BC DE50 FEC0 CD3C 53FE C0CD
3C01 2E1E 08A4 52D1 52FE C0CD 3CFE C0CD
FEC0 CD3C D31F 08A4 50D1 50FE C0CD 3CFE
C0CD 50B0 5001 FEC0 CDA1 A8AD 4350 7050
E96F CF05 FBFE E0CD DF01 A8AD DF01 10A4
DD01 10A4 DD01 D0AD DF01 A8AD 4208 7052
CD00 0000 008F 50FE CCCD 01D0 FED0 CD00
D002 0431 0313 FED8 CDD5 FED8 CD80 50D0
013D 13FE CCCD D543 1E52 FED4 CDD1 5216
A43C FED4 CD00 D0FE C4CD 0A7C C450 FEC8
01FE D4CD C1FE CCCD 00D0 FED0 CDFE D4CD
D00C 1262 B552 22A6 409E 50FE D4CD 24C5
FECC CDD5 0150 D5C7 1105 13FE CCCD D50B
1E52 FED4 CDD1 5216 A43C FED4 CD50 D050
521E A640 9E50 FED4 CD24 C550 D541 1E52
FED4 CDD1 5216 A43C FED4 CD00 80D0 5113
CDC1 FEC8 CD62 5052 1EA6 409E 50FE D0CD
24C5 FED0 CDFE D4CD D017 15FE C8CD 6251
52D1 52FE BECD 3CFE BECD 00B0 C11F 52FE
D4CD D152 16A4 3CFE D4CD 50D0 5001 FED4
7650 D0AD 4262 52FE BECD 3C50 FEC4 CD56
1718 D0AD 4273 52FE BECD 3C01 391E 10A4
00B0 C81F 10A4 50D1 50FE BECD 3CFE BECD
50B0 5001 FEBE CDA1 FEC4 CD85 5050 5050
```

Virtual block number 18 (00000012), 512 (0200) bytes

```
50FE C4CD 5652 FEC4 CD56 0133 1852 5071
520A 80C4 5650 526E 52FE BECD 3CFE BECD
C456 5250 6E50 FEBE CD3C FEBE CD50 B050
01FE BECD A1FE C4CD 5050 5052 7652 5064
FEC8 CDFE C4CD 51FE C4CD 5250 5252 7652
10AC 6452 FEC4 CD56 CE19 5052 7150 0A80
FEC8 CD56 5518 BCDE 1713 FECC CDD5 0150
D500 8A31 0613 FECC CDD5 0C19 8002 01F5
C562 FEC4 CD50 521E A642 9E52 FED0 CD24
C565 5250 5250 7650 5266 520A 7CC4 5650
24C5 013C 1E10 A452 D152 FEBE CD3C FEBE
CD00 B062 01B0 5222 A642 9E52 FED0 CD24
B052 6641 9E50 0CBC DE53 FEBE CD3C 5152
C052 02C4 52FE BECD 3C51 50D0 50FE D0CD
0404 0450 00D0 C51F 10A4 50D1 50FE BECD
3CFE BECD 50B0 5001 FEBE CDE0 A162 6043
F4AD C450 09E2 C240 3C50 F8AD D0F8 AD03
D0F4 AD00 D052 BF1C 0001 049E 5E0C C200
F4AD D0D6 F8AD F4F4 AD50 D050 F4AD C050
6140 9A51 04BC DE50 F8AD D0F4 AD50 D050
0001 049E 54FE 1C00 0504 9EFF 1C4E 4941
4D24 4306 0016 5EFF 38CE 9E00 1C04 0450
DF01 00DB C4DF 0104 50FF FFFF FF8F D008
1350 D5EB 4BCF 01FB 00B3 C4DF 0153 DD1C
50D5 EDE3 CF02 FB09 E2C3 9F01 0A9A C3DF
0180 1C59 5043 5254 5306 0002 FB0A 9AC3
0100 DD01 FF44 CD50 D0E7 AECF 02FB 00E5
C4DF 0100 DD01 0450 FFFF FFFF 8FD0 0813
FF4C CD50 D0E7 8ACF 02FB 00FB C4DF 0100
DD01 FF48 CD50 D0E7 9CCF 02FB 00F0 C4DF
6203 D052 0CA3 9E01 50D5 0A11 6204 D052
0CA3 9E0C 1B01 52D1 5262 3C52 09E8 C33E
```

Virtual block number 19 (00000013), 512 (0200) bytes

```
06C4 DF01 625C D052 10A3 9E5C 0108 A3C1
B962 6CD0 5208 A39E 5C0C A3DE 04A3 03D0
D5EA 7FCF 01FB 0110 C4DF 0162 5CB0 52FF
50CD 9E5C 04A3 A45C 503C E938 CF01 FB01
5254 5306 0002 0200 9102 FB0A 9AC3 DF01
0130 C4DF 0104 50FF FFFF FF8F D008 1350
0450 FFFF FFFF 8FD0 0813 50D5 ED17 CF02
FB09 E2C3 9F01 0A9A C3DF 01C7 1C59 5043
0600 02FB 0AFD C3DF 010A 9AC3 DF01 FF52
CD50 D0E6 E2CF 02FB 013A C4DF 0100 DD01
FFFF FFFF 8FD0 0813 50D5 F216 CF02 FBFF
56CD 9F01 CCAD DF01 801C 5950 4352 5453
5431 031F 09E8 C352 B152 52F7 52FF 63CD
9AFF 63CD 0090 6200 B052 FF5A CD9E 0450
61CD 0090 0105 1D31 031F 09E6 C352 B152
52F7 52FF 62CD 9AFF 62CD 0090 0150 D505
52F7 52FF 60CD 9AFF 60CD 0090 0104 E831
031F 09E4 C352 B152 52F7 52FF 61CD 9AFF
C052 0250 C550 04A3 C4FD E2CF 01FB FF60
CDDF 0101 04B3 3180 031F 09E2 C352 B152
A2AD 3C51 A2AD 3C01 50D5 361E 0CA3 5CD1
5CA2 AD3C A2AD 00B0 C4AD 52D0 52FF 52CD
5CB0 5C01 A2AD A1AC AD41 09EA C34C B05C
50C0 5CA2 AD3C 5004 C450 FF60 CD4C 9A5C
A2AD 3C01 50D5 2B1E 04A3 5CD1 5CA2 AD3C
A2AD 00B0 CD1F 0CA3 5CD1 5CA2 AD3C A2AD
5CA2 AD3C A2AD 5CB0 5C01 A2AD A1A4 AD40
5CB0 5C5C F75C 80C4 BD4C 3C5C A2AD 3C50
B0AA AD5C B05C 5CF7 5C50 3CF5 9DCF 01FB
ACAD DF01 131B 0308 A3D1 D81F 04A3 5CD1
00B0 5C2E A34C 9E5C 18C4 5CA0 AD3C 0150
D522 1E14 A3A0 ADB1 A0AD 00B0 0A98 C300
```

Virtual block number 20 (00000014), 512 (0200) bytes

```
F75C FF5F CD9A FF5F CD00 909A AD00 B0E1
1F14 A3A0 ADB1 A0AD 5CB0 5C01 A0AD A16C
E6C3 5CB1 5C5C F75C FF5E CD9A FF5E CD00
9050 D501 35D1 3103 1F09 E8C3 5CB1 5C5C
3103 8002 0200 1F09 E4C3 5CB1 5C5C F75C
FF5D CD9A FF5D CD00 9001 00FF 3103 1F09
FF5C CDDF 0101 0095 3103 1F09 E2C3 5CB1
5C5C F75C FF5C CD9A FF5C CD00 9001 00CA
1E00 0002 588F 5CD1 5C98 AD3C 98AD 5CB0
5C50 F7F3 D6CF 03FB 90AD DF01 ACAD DF01
FB88 ADDF 01A4 ADDF 0190 ADDF 015C DD01
5CFF 44CD C05C 02C4 5C5C D05C 9AAD 3C42
FB80 88AD DF01 90AD DF01 7E50 7050 C0AD
565C DD01 5C98 AD3C C0AD 5050 F57F CF04
CD9A 9AAD 5CB0 5C5C F75C 035C C15C 9AAD
3CFF 5CCD 5C90 5C01 FF5C CD81 F670 CF05
5CFF 5DCD 9AFF 5DCD 5C90 5C01 FF5D CD81
FF6C 3103 1E09 E2C3 5CB1 5C5C F75C FF5C
5C5C F75C FF5E CD9A FF5E CD5C 905C 01FF
5ECD 81FF 3731 031E 09E4 C35C B15C 5CF7
805C B15C 5CF7 5CFF 5FCD 9AFF 5FCD 5C90
5C01 FF5F CD81 FF02 3103 1E09 E6C3 5CB1
3103 1FFF 50CD 9EAD B19E AD5C B05C 5CF7
5C9C AD3C 9CAD 00B0 FECD 3103 1E09 E8C3
FF7E CDDF 01AC ADDF 015C DD01 5CFF 4CCD
C05C 02C4 5C5C D05C 9CAD 3C01 50D5 00A8
5C5C D05C 9EAD 3C46 1E00 0002 588F 5CD1
5C86 AD3C 86AD 5CB0 5C50 F7F3 50CF 03FB
5050 F467 CF04 FBFF 76CD DF01 A4AD DF01
FF7E CDDF 015C DD01 5CFF 48CD C05C 02C4
F556 CF05 FBFF 76CD DF01 FF7E CDDF 017E
5070 50BC AD56 5CDD 015C 86AD 3CBC AD85
```

Virtual block number 21 (00000015), 512 (0200) bytes

9CAD	5CB0	5C5C	F75C	0CA3	5CC1	5C9C	AD3C
9EAD	5CB0	5C5C	F75C	04A3	5CC1	5C9E	AD3C
74CD	B1FF	74CD	00B0	C8AD	FFFF	FFFF	8FD0
ECBA	CF00	FBFF	5B31	031E	FF50	CD9E	ADB1
8002	0200	1B5C	D55C	6C3C	5C6C	3E5C	2EA3
4C9E	5C18	C45C	FF74	CD3C	017F	1E14	A3FF
CD3C	0150	D53A	114F	1107	1500	003F	808F
6C51	5C28	A34C	9E5C	18C4	5CFF	74CD	3C54
1511	0419	C8AD	D509	1FFF	72CD	FF70	CDB1
FF70	CD6C	B05C	2CA3	4C9E	5C18	C45C	FF74
FF74	CD5C	B05C	01FF	74CD	A1C8	AD5C	D05C
5CD0	5CFF	74CD	3CFF	72CD	FF70	CDB0	50D5
CD3C	0154	1E0C	A35C	D15C	FF6E	CD3C	FF6E
CD00	B069	19C8	ADD5	821F	14A3	FF74	CDB1
3C50	5CD0	5CC8	AD18	C552	FF56	CDC0	5202
C452	5CD0	5C6C	3C5C	FF5A	CD3E	51FF	6E80
FF6E	CDA1	6241	5CB0	5C5C	F75C	6C3C	5C6C
3E5C	18A3	409E	505C	C05C	02C4	5CFF	6ECD
50B8	AD00	0000	008F	5050	D500	8331	AD1F
0CA3	5CD1	5CFF	6ECD	3CFF	6ECD	5CB0	5C01
A4AD	DF01	FF64	CDDF	017E	5070	50B8	AD56
B4AD	DF01	ED1F	CF00	FBB4	AD96	7F4C	188F
CF02	FBD0	ADDF	01A4	ADDF	0104	50FF	FFFF
FF8F	D008	1350	D5F5	53CF	06FB	00DD	8001
5202	C452	5CD0	5C6C	3C5C	FF5A	CD3E	D0AD
DF01	0450	FFFF	FFFF	8FD0	0813	50D5	EDD2
6C3C	5CFF	5ACD	3E04	50FF	FFFF	FF8F	D008
1350	D5EF	DBCF	02FB	52DD	0152	FF56	CDC0
FF60	CD9A	FF60	CD5C	905C	01FF	60CD	8162
5CB0	52FF	5ACD	9E5C	5CF7	5C0C	A35C	C15C
F752	FF61	CD9A	FF61	CD52	9052	01FF	61CD
81FB	4E31	031E	09E2	C35C	B15C	5C85	F75C

Virtual block number 22 (00000016), 512 (0200) bytes

```
B152 52F7 52FF 62CD 9AFF 62CD 5290 5201
FF62 CD81 FB19 3103 1E09 E4C3 52B1 5252
E8C3 52B1 5252 F752 FF63 CD9A FF63 CD52
9052 01FF 63CD 81FA E431 031E 09E6 C352
0200 FBFF 56CD DD01 CCAD DD01 0AFD C3DF
010C A3DD 0109 E2C3 9F01 FAAF 3103 1E09
0015 2000 0600 BE0F 0000 0090 00BF 0639
5F6C 6163 6F6C 07EE 1B00 0014 9000 0605
6C08 ED1B 0000 15A4 0006 00BE 0F00 0000
8400 BF06 3031 5F6C 6163 6F6C 08ED 1B00
5F6C 6163 6F6C 08ED 1B00 0016 7000 0600
BE0F 0000 00CA 00BF 0632 315F 6C61 636F
00BF 0635 315F 6C61 636F 6C08 ED1B 0000
1780 0006 00BE 0F00 0001 1000 BF06 3331
0F00 0004 6C00 BF06 3631 5F6C 6163 6F6C
08ED 1B00 0018 E000 0600 BE0F 0000 015E
9000 0600 BE0B 0000 0044 00BF 0639 315F
6C61 636F 6C08 ED1B 0000 1D4C 0006 00BE
FB80 1B00 0000 1800 0610 0171 09B9 0900
0007 0A00 BF06 6E69 616D 04EE 1B00 001D
FAFA 0000 F8F4 0402 FF02 02F5 CDFB 0302
FF02 02FB E7F2 F7EE E400 E1FB 0502 00B9
0602 FF02 02FD FC00 FCBC 0004 02FE 00FC
EE00 DC00 0102 FDFD F700 EFF3 F3F6 0000
0102 ED01 0200 F601 0200 0102 F9F0 EB00
0102 F1F7 F800 0102 FCDF 0102 D400 FAFB
8002 FF02 02FA E901 02B5 F2C7 0102 CE01
02EA 0102 DCA7 D900 0102 EEDB 0102 DBED
E1E1 E100 F2F8 0102 EBEE ED01 02F4 0402
FC28 02E6 D5E8 E500 E7B5 F201 02F1 F906
F2F8 0102 EBEE ED01 02F4 0402 FFFA 0000
D6C3 D600 FAE5 D1EF 0002 02F2 00EA DE00
```

Virtual block number 23 (00000017), 512 (0200) bytes

EC00	FCF3	E9ED	00F2	ED01	02F8	0102	F405
02FE	00F8	E4E4	E4FC	00F2	DB00	DBDB	DB00
02FF	B9FC	0302	DCE2	0000	FCE5	00EE	00FC
F8C7	D500	EC00	ECFE	DF00	EF00	D4EC	00D8
CD01	02FB	DC01	02F5	0102	F406	02FF	0302
FAF5	CE01	02FC	FCF6	D801	02F6	F5FB	05E1
02FA	F201	02C1	ECEF	EEE9	0102	E7BB	EDBB
EDFB	0001	02E9	EC01	02EC	F3E7	8005	01EE
D000	0102	D5F8	F401	02FF	0302	F7F3	CFF4
0202	FFE4	E6E9	FBD6	D900	FCF4	0302	FF03
FB02	02FC	0602	CFED	0102	E609	02EA	0000
00F0	00FB	DA00	EBBD	00F0	F3F0	00E7	FC00
F2E1	B895	8E9C	00D3	00DF	F701	02F6	EB04
02FF	0202	FCD8	0102	CCFC	ED02	02FF	FCE2
0001	02F4	FC01	02F4	0202	FF02	02FB	EAEA
E298	F8FC	0001	02EC	F403	0280	FE02	02FA
02F4	0202	FF0B	02FB	F0F1	F1E1	EB00	0102
F4FC	0102	F402	02FF	0202	FBF0	F3F1	E1DD
FB04	02F8	00D3	EDE7	D200	0102	EF00	0102
E3C2	F8F4	0502	FD02	02EA	E7E3	E3B4	F801
0402	FD02	02FF	F0DC	FFFE	FA00	E900	0102
F2F0	0102	BCB9	EFEC	D201	02F4	0702	FFB9
F7E1	B8C1	008B	01FB	0102	0004	0200	0102
FC00	C9D3	D7DD	0001	02AF	8000	0302	F6EB
E900	FA00	0102	DAA0	0102	B7E9	0102	EEEE
D900	FBF9	0001	02DA	F901	02F7	F600	0102
1002	FF02	02FB	FCEC	EE00	FCFC	0102	F403
02FF	0202	FC00	0000	00C5	EFF2	F000	0102
E8E6	0001	0200	F8E6	0000	EFEE	E5EF	EA00
00E7	F4F5	FCF4	ECEE	EEEE	E5EF	EA00	00E5
DA00	0202	E8E8	E8E7	FC00	00D3	FB00	0102
00E7	C7BC	E500	0102	0097	0502	E800	E800

Virtual block number 24 (00000018), 512 (0200) bytes

```
E800 DBF8 0002 02FB 00D5 00E3 D700 0102
CC00 0202 E300 8301 0202 00E5 D900 0102
D8B9 10FB F001 0200 0202 00FB 00EA C300
EF00 0102 FBED 0000 00F4 F900 EEEE 00FB
0008 0001 4545 5246 0400 0800 0101 0151
BD01 050E FCD8 C6E3 E3E8 00D5 E700 0102
5845 0400 0800 0135 5458 4504 0008 0001
3654 5845 0400 0800 0143 4F4C 4C41 4D06
0800 0131 5458 4504 0008 0001 3254 5845
0400 0800 0133 5458 4504 0008 0001 3454
0500 0800 0134 3154 5845 0500 0800 0135
3154 5845 0500 0800 0136 3154 5845 0500
5845 0400 0800 0132 3154 5845 0500 0800
0131 3154 5845 0500 0800 0133 3154 5845
0137 3154 5845 0500 0800 0139 5458 4504
0008 0001 3031 5458 4505 0008 0001 3854
001C 0000 1D90 0000 0A00 0259 5043 5254
5306 0008 0001 434F 4C4C 414D 0600 0800
0200 4544 4F43 2405 0000 249A 00E9 0200
4E49 414D 2443 0600 0800 014E 4941 4D04
0004 019D 0200 4E49 4454 5305 0000 0004
019D 0200 4154 4144 2405 0000 0B60 0189
0000 0145 0189 0200 5252 4544 5453 0600
0000 0401 9D02 0054 554F 4454 5306 0000
0300 0001 4401 BD02 0053 544E 4154 534E
4F43 5F47 4E49 5254 535F 5241 4843 2416
0450 01D0 0450 00D0 0450 FFFF FFFF 8FD0
0813 50D5 E6A7 CF05 E702 0027 0153 414D
B001 50D5 00D1 FFFF 0000 001D 9000 0003
0008 5800 1500 2900 29FA 500A 0B00 0504
FB52 DEAD 3C50 D547 1850 E0AD D1F2 1C31
3154 5845 0500 00FB E0AD 00D0 DEAD F4AD
```

APPENDIX G

Dump of file [YOAV_B.COLOR.COMPUTER.PATENT]PAT2.OBJ;2
on 18-NOV-1990 File ID (11403,7,2)
End of file block 14 / Allocated 15

Virtual block number 1 (00000001), 512 (0200) bytes

3A383120 30393931 2D564F4E 2D383130
2E315604 32544150 04020000 00000031
33562043 20584156 01000010 00000000
00000000 00000000 00000000 00003831
69614620 0AD55000 05040001 00048081
3C80F850 00010402 02003135 302D312E
6F432072 6F662073 74616F6C 66206425
20657461 636F6C6C 61206F74 2064656C
50000004 00000001 00000008 00000008
00000008 F0502000 01055D64 255B6665
B5F6AD50 B05050F7 5001C250 04A232F8
AD00D052 A11C0001 049E5E0C C20004FA
04BCDE51 F6AD32F8 AD50D050 F8ADC450
50D05010 A2403250 F6AD3201 3A19F6AD
CDD0C718 F6ADB5F6 AD50B050 F6AD01A3
F8AD50D0 50F8ADC0 5050D050 60413250
F4AD01D0 F8AD00D0 53A41C00 01049E54
FE1C0005 049E5E10 C2001C04 0450FFF8
AD5CD05C F4ADC45C 5CD05C10 A34C325C
F8ADD001 50D52A1E 52F8ADD1 5204A332
52F8ADD1 5204A332 F8AD00D0 D91F5CF8
ADD15C04 A332F8AD 5CD05C01 F8ADC1F4
4250D0E2 1C434F4C 4C414D06 0001FB5C
DD015C04 F4ADC552 F8ADD001 50D5541E
49525006 0003FB64 DF01F4AD DD01F8AD
DD011A12 1FF0C34C D55CF8AD D01FF0C3
AF1F5CF8 ADD15C04 A332F8AD 5CD05C01
F8ADC104 50FFFFFF FF8FD080 1C46544E
AE310319 10AC50D1 50FE12CD 32FE12CD
00B05EFE 0CCE9E00 7C040450 00008F32
FE14CD00 0000008F 500150D5 01823103
15FE12CD FE10CDB1 FE10CD00 F750D501

Virtual block number 2 (00000002), 512 (0200) bytes

54FE14CD	56015418	FE10CDFE	0ECDB1FE
0ECD00B0	00AB3103	13FE10CD	FE12CDB1
32510BC4	51FE10CD	3252FE18	CD405650
51C050FE	0ECD3251	0BC4DE51	FE10CD32
A1FE14CD	52505254	76545260	52506450
FE18CD40	A4020200	565051C0	50FE0ECD
56FE10CD	32520BC4	52FE10CD	32AD19FE
10CDFE0E	CDB1FE0E	CD52B052	01FE0ECD
52FE14CD	56546243	565204BC	DE5352C0
53FE10CD	32520BC4	52FE10CD	325652C0
B00150D5	00B831FE	18CD4652	50525076
801C5452	51530400	02FB7E54	70545262
FE0ECD32	530BC453	FE12CD32	50FE14CD
5650D555	18FE10CD	FE0ECDB1	FE0ECD00
52FE18CD	42565253	C052FE0E	CD32530B
C453FE10	CD3254FE	18CD4256	5253C052
FE10CDFE	0ECDB1FE	0ECD52B0	5201FE0E
CDA1FE14	CD525052	50765054	60545264
53FE10CD	32520BC4	52FE12CD	32545280
C054FE10	CD32520B	C452FE12	CD32AD19
52FE10CD	32530BC4	53FE10CD	32555262
52FE14CD	56556243	565204BC	DE5352C0
A1FE81FE	10CD01FE	12CD3DFE	18CD4452
50525576	55526652	FE18CD42	565253C0
12CD32FE	12CD00B0	FE543103	1810AC52
D152FE12	CD32FE12	CD52B052	01FE12CD
CDFE10CD	B1FE10CD	00B0FE14	CD008000
00008F50	0100BA31	031910AC	52D152FE
32530BC4	53FE12CD	32546342	56530CBC
DE52FE10	CD3250FE	14CD5601	4B18FE12
01FE10CD	A1FE14CD	52505250	76505460
54526452	FE18CD42	565253C0	52FE10CD

Virtual block number 3 (00000003), 512 (0200) bytes


```
12CD32FE 12CD52B0 5201FE12 CDA16044
52508002 02005255 76555266 52FE18CD
C3310318 FE12CDB5 FE12CD52 B05252F7
5210AC01 C3FF4731 031810AC 52D152FE
52FE10CD 32FE10CD 52B05252 F75201C0
52FE12CD 32FE14CD 00000000 8F500100
530BC453 FE10CD32 54634256 530CBCDE
52FE10CD 3250FE14 CD564C18 10AC52D1
10CDA1FE 14CD5250 52507650 54605452
648052FE 18CD4256 5253C052 FE12CD32
FE12CD32 500CBCDE 54FE12CD 32B41910
AC52D152 FE10CD32 FE10CD52 B05201FE
53C052FE 12CD3253 0BC453FE 12CD3255
526252FE 14CD5655 62435652 0CBCDE53
FE12CDB5 FE12CD52 B052FE12 CD01A360
44525052 55765552 6652FE18 CD425652
70111850 04AC7150 60415650 0CBCDE51
8010AC32 5E0CC200 FC0104FF 3E310319
56500CBC DE5150C0 5014AC32 5110AC32
50D501BD 31045000 00000000 0000008F
AC323E12 0114ACB1 01933104 50000000
00000000 008F700F 1804AC50 71506041
51D15001 C25018AC 325110AC 32291250
04AC7150 60415650 0CBCDE51 01C05110
14ACB101 01503104 50000000 00000085
00008F70 0150D50F 11045008 70091250
5150C050 14AC3251 10AC3253 0CBCDE52
10AC3201 50D50142 31045008 700A1201
```

Virtual block number 4 (00000004), 512 (0200) bytes

```
325404AC 52635262 4356520C BCDE5310
AC326B18 60416342 51500CBC DE5101C2
62435652 0CBCDE53 10AC3250 62435652
0CBCDE53 01C25352 C05214AC 325310AC
5210AC32 52DD0152 01C25214 AC3252DD
80020200 015218AC 32545066 50526252
008F5001 0911F8AD 52505254 76545064
FEDDCF06 FB7E04AC 700CACDD 0152DD01
51520CBC DE5352C0 5214AC32 5310AC32
550CBCDE 5401C054 10AC32F8 AD000000
10AC3256 04AC6256 62435652 0CBCDE53
52C05214 AC325310 AC327318 62436544
52624356 520CBCDE 5301C053 10AC3298
54624356 520CBCDE 5352C052 14AC3253
DD015201 C05210AC 3252DD01 5201C252
14AC3252 DD015218 AC325654 66545262
00000000 8F500811 F4AD5250 52567656
5064FE40 CF06FB7E 04AC700C ACDD0152
F757801C 0001049E 5EB8AE9E 00FC0404
50547054 526052F4 AD5654F8 AD56F4AD
C6AD00B0 FFEAC6AD 01033D18 A74310A7
42B052C6 AD3253C6 AD32FFFF FFC6ED00
00008F50 5352C053 CAAD3252 0EC452C6
AD32CAAD 00F70150 D5541804 A7C6ADB1
505354C0 5352C052 CAAD3253 18A74232
52C6AD32 540EC454 C6AD3220 A7430000
C680AD00 B0AF1904 A7C6ADB1 C6AD52B0
5201C6AD A1FFC3CA AD01023D 20A74308
AD32520E C452C6AD 325F1818 A742CAAD
B152C6AD 32CAAD03 B07F1804 A7C6ADB1
53C6AD32 550100C7 42565253 C05202C2
52CAAD32 530BC453 C6AD3254 52C054CA
```

Virtual block number 5 (00000005), 512 (0200) bytes

```
A7445250 52557655 10665552 60520100
C7425652 53C05201 C252CAAD 32530BC4
F2C6AD52 B05201C6 ADA1A119 18A742CA
ADB152C6 AD32CAAD 52B05201 CAADA120
CAAD00B0 0150D500 CA310319 04A78002
0200C6AD B1C6AD00 B0811904 A7C6ADB1
781810A7 42C8ADB1 52C6AD32 C8AD00B0
0100A131 031918A7 42CAADB1 52C6AD32
52C6AD32 5452C054 C8AD3252 53C05221
C452CAAD 32530000 016B8FC4 53C6AD32
AD3220A7 42DF0152 0EC452C6 AD3252DD
0152CAAD 3203DD01 52DD0152 18A74232
52505250 76FC7ACF 06FB7E52 70805201
00C74256 5253C052 C8AD3253 0BC453C6
52B05201 CAADA188 1910A742 C8ADB152
C6AD32C8 AD52B052 01C8ADA1 0940C744
1804A7C6 ADB1C6AD 52B05201 C6ADA1FF
60310318 18A742CA ADB152C6 AD32CAAD
CAADB152 C6AD32CA AD00B001 50D50124
31031904 A7C6ADB1 C6AD00B0 FF393103
00000000 8F5050D5 00D43103 8015CAAD
C8ADB1C8 AD00F701 00FB3103 1918A742
0000016B 8FC452C6 AD3250F4 AD566A18
10A742BC ADB152C6 AD32BCAD 00B0F4AD
016B8FC4 52C6AD32 540940C7 42565253
C052BCAD 325352C0 5321C453 CAAD3252
50546054 52645209 40C74256 5253C052
BCAD3253 52C05321 C453C8AD 32520000
C6AD3296 1910A742 BCADB185 52C6AD32
BCAD52B0 5201BCAD A1F4AD52 50525076
798FC452 C6AD3254 52C054C8 AD325253
C0520BC4 52CAAD32 53000000 798FC453
```

Virtual block number 6 (00000006), 512 (0200) bytes

```
B0C74250 01B0C742 F4AD5052 53C052CA
AD325352 C0530BC4 53C8AD32 52000000
18A742CA ADB152C6 AD32CAAD 52B05201
CAADA1FF 2EC8AD01 CAAD3D01 B0C74401
C4AD00B0 FEDF3103 1804A7C6 80020200
ADB1C6AD 52B05201 C6ADA1FF 06310318
54310319 1CA7C2AD B1C2AD00 B0010276
3103191E A7C4ADB1 F8AD52D0 52C4AD32
0E310319 18A7BEAD B1BEAD00 B050D502
31310319 1AA7C0AD B1C0AD00 B050D502
01C6ADA1 D0AD4000 0000008F 5050C6AD
32011E18 04A7C6AD B1C6AD00 B050D502
031952E4 ADD15216 A732E080 ADE4ADD0
E4AD00D0 E31904A7 C6ADB1C6 AD50B050
5012A732 ECAD00D0 0150D501 53310319
50E8ADD1 5014A732 E8AD00D0 01017B31
0150D500 FF310319 50F0ADD1 5010A732
F0AD00D0 0150D501 29310319 50ECADD1
50527652 50645009 40C74056 50F0AD50
C15021C4 50BEAD32 52CCAD56 CCAD6750
52765250 64500EEC C7408056 50ECAD50
C15021C4 50C0AD32 52CCAD56 CCAD5050
52765250 64501498 C7405650 E8AD50C1
5021C450 C2AD3252 CCAD56CC AD505050
501A44C7 405650E4 AD50C150 21C450C4
AD3252CC AD562013 06A7B5CC AD505050
BAAD3201 50D54118 04A7BAAD B1BAAD00
B04C13CC AD53CCAD 50505052 76525064
50645050 56506041 4D855004 BCDE51E0
ADC051BA AD3253CC AD5655D0 AD425652
01F0ADC1 C21904A7 BAADB1BA AD50B050
01BAADA1 D0AD4250 50505576 55536053
```

Virtual block number 7 (00000007), 512 (0200) bytes

ADC1FF04	31031850	F0ADD150	10A732E0
AD50D050	E0ADC050	04A732F0	AD50D050
14A732E8	AD50D050	01E8ADC1	FEDA3103
1850ECAD	D15012A7	32ECAD50	D05001EC
1850E4AD	D15016A7	32E48002	0200AD50
D05001E4	ADC1FEB0	31031850	E8ADD150
521FF0C7	42D053F8	ADD052C6	AD320150
D52B1804	A7C6ADB1	C6AD00B0	FE863103
B05201BE	ADA1D819	04A7C6AD	B1C6AD50
B05001C6	ADA16243	D0AD4050	50C6AD32
B1C0AD52	B05201C0	ADA1FDF4	31031818
A7BEADB1	F8AD52D0	5201F8AD	C1BEAD52
C4ADA1FD	AE310318	1CDCA7C2	ADB1C2AD
52B05201	C2ADA1FD	D1310318	1AA7C0AD
801C0001	049E5EFF	58CE9E00	0450D504
FD8B3103	181EA7C4	ADB1C4AD	52B05201
031904A2	FF74CDB1	FF74CD00	B050D502
3C310319	04A2FF76	CDB1FF76	CD00B052
01033DFF	64CD4018	A24CB05C	FF62CD32
50FF62CD	32FFFFFF	62ED008F	99021431
C0310319	FF6ACDFF	72CDB1FF	72CD00B0
FF64CD4C	01B05CFF	74CD32FF	E6FF62CD
FF6ECDB1	FF6ECD00	B00150D5	01963103
19FF68CD	FF70CDB1	FF70CD00	B050D501
0150D501	40310319	FF64CDFF	6CCDB1FF
6CCD00B0	0150D501	6B310319	FF66CD80
40FF6CCD	4CB05CFF	62CD3250	FF62CD32
0150D529	1804A2FF	62CDB1FF	62CD00B0
5CFF74CD	32FF60CD	00B0DA19	04A2FF62
CDB1FF62	CD5CB05C	01FF62CD	A1FF58CD
DF015CFF	60CD32FF	58CD4CFF	60CDB05C
FF74CD32	0150D560	1818A24C	FF60CDB1

Virtual block number 8 (00000008), 512 (0200) bytes

```
5CFF76CD 3250FF60 CD3251FF 60CD326C
50D05CD0 AD4CDEF2 D0CF01FB FF58AECD
FF74CD32 FF60CD5C B05C01FF 60CDA1A4
AD416C40 505C1FF0 C24CD050 D0AD40D0
FF78CDDF 015CDD01 80020200 5C18A24C
325CFF74 CD32A319 18A24CFF 60CDB15C
FF60CD00 B0F37ECF 04FB01B0 C24CDF01
5C000000 798FC45C FF74CD32 A4ADDF01
51D0AD40 D05CFF76 CD3250FF 60CD3201
50D53D18 18A24CFF 60CDB15C FF74CD32
74CD32FF 60CD5CB0 5C01FF60 CDA16041
FF78CD4C 505CFF60 CD32501F F0C24CD0
0318FF64 CDFF6C80 CDB1FF6C CD5CB05C
01FF6CCD A1C61918 A24CFF60 CDB15CFF
5C01FF70 CDA1FE98 310318FF 66CDFF6E
CDB1FF6E CD5CB05C 01FF6ECD A1FEC331
FF72CDB1 FF72CD5C B05C01FF 72CDA1FE
6D310318 FF68CDFF 70CDB1FF 70CD5CB0
CDA1FDEC 31031804 A2FF74CD B1FF74CD
5CB05C01 FF74CDA1 FE423103 18FF6ACD
0001049E 5E08FDC2 003C0104 FDC63103
1804A2FF 76CDB1FF 76CD5CB0 5C01FF76
150B10A5 43B110A5 4352B052 2000C542
F752FAAD 3253FAAD 32FAAD00 F755801C
FB06A500 B004A503 B0081201 200CC5D1
FFD7FAAD 01033D04 50FFFFFF FF8FD008
32F8AD00 B0661804 A5FAADB1 FAAD00B0
0450FFFF FFFF8FD0 081350D5 F19FCF00
500BC450 FAAD3254 5CC054F8 AD325C0B
C45CFAAD 32461810 A542F8AD B152FAAD
F8ADA101 00C5445C 505C5276 52506450
65565220 10C54C6E 5C50C05C F8A5AD32
```

Virtual block number 9 (00000009), 512 (0200) bytes

04A5FAAD	B1FAAD52	B05201FA	ADA1BA19
10A54CF8	ADB15CFA	AD32F8AD	5CB05C01
01049E5E	18C2001C	04045000	008F32FC
CBCF00FB	F724CF01	FB20C0C5	DD019A19
52B05252	F7528002	02000120	C4C442C1
52EAAD32	53EAAD32	EAAD00F7	54ED1C00
52EAAD32	EEAD00B0	00F83103	1904A4EA
ADB1EAAD	00B0FFE2	EAAD0103	3D08A443
0100A631	031908A4	42ECADB1	52EAAD32
ECAD00B0	0100D231	031918A4	42EEADB1
AD71F0AD	52705250	64506456	5220D4C4
4C565C50	C05CEAAD	325004C4	50ECAD32
AD32F0AD	00800000	00000000	008F700C
18F0AD73	50D51311	F0AD0870	081508F0
4C325CEA	AD32525C	C052ECAD	325C50C0
5C21C45C	EEAD3250	0000016B	8FC450EA
7EF0AD70	20A44CDF	015C0EC4	5CEAAD32
5CDD015C	EEAD3203	DD015CDD	015C18A4
A44CECAD	B15CEAAD	32ECAD5C	B05C01EC
ADA10940	C4425C50	5C5076F4	4CCF06FB
A1FF2F31	E0031818	A442EEAD	B152EAAD
32EEAD52	B05201EE	ADA1FF5B	31031808
1C000104	9E5E3CC2	00FC0150	D504FF08
31031804	A4EAADB1	EAAD52B0	5201EAAD
52E4ADD1	5204A732	E4AD00D0	DCAD52D0
5253D053	52C45218	A732531A	A7325780
52E8AD52	C152E4AD	21C5012D	1852E8AD
D15208A7	423252E4	ADD0E8AD	00D05318
52E8ADD1	5208A742	3252E4AD	D0E8AD52
D05201E8	ADC122E8	C7420000	00008F50
A732E0AD	ECADD0EC	AD00D0AD	1952E4AD
D15204A7	32E4AD52	D05201E4	ADC1D419

Virtual block number 10 (0000000A), 512 (0200) bytes

```
52C05322 E7C79852 ECAD21C5 2F1306A7
B50150D5 01F53103 1952ECAD D152AA1E
E0AD52D0 52E0ADC0 52DCADC4 5252D052
1CA73213 12D8AD53 D8AD1A44 C7435053
12D8AD53 80020200 01933103 1952F0AD
D1521CA7 32F0AD00 D0D8AD08 50010511
52645214 98C74356 5352C053 22E6C798
52F0AD21 C554D8AD 560150D5 018B3103
32F4AD00 D0010137 31E0AD52 D052DCAD
E0ADC10E 12D4AD53 D4AD5250 52547654
5352C053 22E5C798 52F4AD21 C554D4AD
560150D5 01253103 1952F4AD D1521AA7
E0ADC080 5252D052 18A73212 12D0AD53
D0AD5250 52547654 5264520E ECC74356
C6AD00B0 0150D500 C3310319 52F8ADD1
5218A732 F8AD00D0 00D431E0 AD52D052
40C74356 5352C053 C6AD3252 F8AD21C5
54D0AD56 50D50093 31031908 A7C6ADB1
014B1852 C8ADD152 04A732C8 AD00D059
13CCAD53 CCAD5250 52547654 52645209
1FF080C7 4CD050E0 ADD05CC8 ADD05522
E8C74456 545CC054 C6AD325C C8AD21C5
AD5CD05C 01C8ADC1 22E8C744 5C505C55
76555260 52506450 CCAD5652 6C40565C
ADC1FF6F 31031808 A7C6ADB1 C6AD52B0
5201C6AD A1B6195C C8ADD15C 04A732C8
F4ADC1FF 40310318 52F8ADD1 5218A732
E0AD52D0 5201E0AD C1F8AD52 D05201F8
1C85A732 F0AD52D0 5201F0AD C1FEDE31
031852F4 ADD1521A A732F4AD 52D05201
D1521EA7 32ECAD52 D05201EC ADC10150
D5FE7B31 0613D8AD 530B1852 F0ADD152
```

Virtual block number 11 (0000000B), 512 (0200) bytes

423252E4 ADD0E8AD 00D06318 52E4ADD1
5204A732 E4AD00D0 FE0E3103 1852ECAD
0000447F 8F645422 E8C74356 53E8AD52
C152E4AD 21C5013D 1852E8AD D15208A7
3252DA02 0200E4AD D0E8AD52 D05201E8
ADC122E8 C7435250 52547654 00000000
000C049D 1952E4AD D15204A7 32E4AD52
D05201E4 ADC1C419 52E8ADD1 5208A742
0153F8AD D0FFF8CD 008F9852 EF1C0001
049EFF1C 4E49414D 24430600 165E28C2
43E4ADD1 53F8ADD0 E4AD00D0 2000C243
50D0D31C 30325458 45050001 FBF8ADDD
54584505 0002FBF8 ADDD01E4 ADDD015C
E4AD5CC1 5CF8AD0B C550D537 182000C2
CB192000 C24CE4AD D15CF8AD D0E4AD5C
D05C01E4 ADC12010 C24C50D0 E21C3132
D0FFF8CD 00D08DF8 AD03F3FA 66CF00FB
20C0C250 D0E21C32 32545845 050000FB
00D0635C D05320C4 C243DE5C 50D0CF1C
33325458 45050001 FBF8ADDD 0153F8AD
ADDD01E0 ADDD0153 E0AD5CC1 5CF8AD04
C5381E20 C4C243E0 ADD153F8 ADD0E0AD
ADD0E0AD 5CD05C01 E0ADC120 D4C2435C
505C504E 8C1C3432 54584505 0002FBF8
1F20C8C2 ECADD1EC AD00D0FA CECF00FB
97F8AD03 F3C81F20 C4C24CE0 ADD15CF8
D0C2F4AD D1F4AD00 D001009E 31031F20
CCC2F0AD D1F0AD00 D00150D5 00C43103
00FB22E7 C25C905C F4ADF622 E6C25C90
5CF0ADF6 22E5C25C 905CECAD F67D1E20
AD21C5F8 AD00D050 D53E1E20 C4C2E8AD
D1E8AD00 D0D11C38 324C4143 4F4C0700

Virtual block number 12 (0000000C), 512 (0200) bytes

```
32545845 050004FB E8ADDF01 F8ADDD01
7E507050 22E8C24C 565CE8AD 5CC15CF8
5CD05C01 F4ADC1C4 1F20C4C2 E8ADD1E8
AD5CD05C 01E8ADC1 D9F8AD03 F3C71C37
03050200 31031E20 CCC2F0AD D1F0AD5C
D05C01F0 ADC1831F 20D0C2F4 ADD1F4AD
6E6905F0 1B000000 00000600 BE0C3254
41500400 00000700 BC0BF152 000001FF
66656F63 5F636F6C 6C610AEB 1B000000
64000600 BE110000 006400BF 06666F64
036F00BF 06796B73 656C6F68 4308ED1B
0000011C 000600BE 0F000000 B800BF06
70000600 BE0F0000 01E400BF 06626C61
764505F0 1B000004 8C000600 BE0C0000
EA1B0000 0C1C0006 00BE1200 0005AA00
BF067374 6F6E4B74 655308ED 1B000006
6C06EF1B 00000E78 000600BE 0D000002
5B00BF06 65726175 71537473 61654C0B
0635326C 61636F6C 07EE1B00 000F5800
0600BE0E 000000E0 00BF0631 6C61636F
000002FC 00BF0638 326C6163 6F6C07EE
1B000010 90000600 BE0E0000 013500BF
00000610 01E909B9 09000001 D600BF06
6E69616D 04EE1B00 00138C00 0600BE0B
00F8EE00 F5E700EF C4ED0202 FF0202FA
F2E9EC00 E8F40402 00B9FB80 1B000000
0302C2A7 F700ECDD 030200AE 9D00FA00
BD9E00F4 F700ECEB F90802FF 0E02FAED
009201F5 00890100 0202FAF2 00EFEC00
E2D6E2FB 0302FE0F 02ECC299 F700E9E7
02EF00A3 010102F0 ECEF8103 02F5E903
02E1E900 FCF2E303 02F30702 FF00F1F6
```

Virtual block number 13 (0000000D), 512 (0200) bytes

```
00ECECEC E9D70102 000302F0 F0F0E9ED
E8F8B886 F8000202 F000A801 02EC0001
F50102DE 000202F0 000102EE F20102FD
0902ABCA 9C00B400 0102FBDB E0E0E0FC
00850202 FE0602EA EAA4B0D3 0102E7E0
E90102F3 00EACA01 02000402 ECECECED
DE0102F4 0102FF04 02FAFBF6 F00102AA
000202F5 EF010200 FCFC00F2 DAF48005
E8EC00FB 000302E7 AD0102F2 EE0102F4
0102FC04 02BDB0EB B95BF6E2 00ECECF2
01029DF2 EAE2EAE1 ED00A700 0102FBE0
00F00001 02E9ECDF 00E8F1DF 00E4F900
C7C4F2F9 F7F7F700 F4F0EEFB FBB7E500
0102FBFB FBF4B8EB 000102FA EE0102FF
00015452 51530400 08000138 324C4143
4F4C0700 08000101 00FD00BD 01FE050E
00013732 54584505 00080001 434F4C4C
414D0600 08000146 544E4952 50060008
00080001 32325458 45050008 00013332
54584505 00080001 34325458 45050008
41434F4C 0700FC00 00109000 000A0002
30325458 45050008 00013132 54584505
02004E49 414D2443 06000800 014E4941
4D04000C 0000138C 00000A00 0238324C
0004019D 02004154 41442405 000024F8
01890200 45444F43 24050000 156200E9
06000000 04019D02 0054554F 44545306
00000004 019D0200 4E494454 53050000
534E4F43 5F474E49 5254535F 52414843
24160000 002C0189 02005252 45445453
FF3F3103 1E20C8C2 ECADD1EC AD53D053
01ECADC1 FF63E502 001D0053 544E4154
```

Virtual block number 14 (0000000E), 512 (0200) bytes

```
00000000 00000000 00000000 0000FFFF
00000013 8C000003 00080004 045001D0
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
```

Appendix H

Annexes F and G comprise the object code of files called 'pat1.obj' and 'pat2.obj', respectively. This Annex comprises the source code for a file called 'patdum.c'.

'patdum.c' explains and demonstrates how a calling program to 'pat1.obj' and 'pat2.obj' should be written. 'patdum.c' is written in C, but is mostly composed of comments, found between #ifdef and #endif. Any C programmer will understand from it how to write a calling program according to his needs. Parameters that are left to the programmer's control include the dimension of the input and output digital representation, the number of patches measured and the structure of the grid. The programmer needs to supply, using his version of 'patdum.c', the measured data (Tables 30 and 32 in Fig. 1) and the programmer needs to use the program to receive the results (Table 36).

'patdum.c' should be compiled using the C compiler on the VMS operating system on the VAX computer manufactured by Digital Equipment Corporation. This will produce an object code file that is usually called 'patdum.obj'. Assuming that the compiler is invoked by the command, 'cc', then the full command line is typically:

```
cc patdum
```

'pat1.obj', 'pat2.obj' 'patdum.obj' and a library should be linked together on VMS using a linker that is typically available on the VAX. The library is the standard library provided for C programs. Assuming that the linker is invoked by 'link', and that the library is called 'c$1', the full command

line is typically:

link pat1,pat2,patdum,c$1/opt

This produces an executable file.

'patdum.c' is composed of C routines, all called ext..(). Routines ext1() to ext6() present output from pat1.obj to the outside world, and ext8() to ext17() take input from the outside world and present it to pat1.obj. Routines ext20() through ext22() give input to pat2.obj and ext23(), ext24() and ext27() take output from pat2.obj and present it to the outside world. The programmer is free to leave the output routines as they are but most provide useful input routines.

```
/*
 *
 *
 *                        APPENDIX H                            Page 1
 *
 *
 *************************************************
 *
 *
 *                      PRISMA  Prepress System
 *
 * Copyright (c) 1985, SCITEX Corporation Ltd.
 *                      All Rights Reserved
 *
 ***********************************************************
 * Module name:          patdum ( for patent application )
 * Programmer's name:    Yoav Bresler
 * Date:                 18-Nov-1990
 * Language:             c
 * Where to find it:     scc; sd [.patent]
 *
 *      The following is a descrption of the interface to
 *      'pat1' and 'pat2'.
 *      Examples are provided within, but the whole is not
 *      a working example of such an interface.
 ************************************************************
 */
#ifdef  COMMENTS_NOT_CODE
        COMMENTS_NOT_CODE shouldn't be defined,
        so text between 'ifdef'  and 'endif' is comments.
#endif
```

```
 *
 *                                                                   Page 2
 *
 *
 *
 *
 *
 ****************** Interface to PAT1 ******************
void ext1(i)    int i;
{
#ifdef  COMMENTS_NOT_CODE
        Let it be known that 'i' is the dimension of the
        digital representation
        which is the output of the resulting Table 36.
        Example 1: i=3 for cyan,magenta,yellow
        Example 2: i=4 for cyan,magenta,yellow,black
#endif
};
void ext2(i)    int i;
{
#ifdef  COMMENTS_NOT_CODE
        Let it be known that 'i' is the dimension of the
        digital representation
        which is the input of the resulting Table 36.
Example 1: i=3 for cyan,magenta,yellow
Example 2: i=4 for cyan,magenta,yellow,black
#endif
}
void ext3(i,j)  int i,j;
{
#ifdef  COMMENTS_NOT_CODE
        Let it it be known the the grid of Table 36,
        in the 'i'-th input dimension is define by 'j' numbers.
        'i' is in the range 0 to 3
Example 1: if the cyan grid is
        (0,10,20,30,40,50,60,70,80,90,100)
        then j=11 for i=0.
#endif
}
void ext4(i)    int i;
{
#ifdef  COMMENTS_NOT_CODE
        Let it be known that 'i' is the maximum numbers
        defining a grid in any of the input dimensions.
#endif
}
void ext5(i,j,k)        int i,j,k;
{
#ifdef  COMMENTS_NOT_CODE
        Let it be known that 'k' is the number at the
        'j'-th grid stop at the 'i'-th input dimension.
Example 1: if the cyan grid is
        (0,10,20,30,40,50,60,70,80,90,100)
        than k=90 for i=0 and j=9.
#endif
}
/*
```

```
         *
         *
         *                                                   Page 3
         *
         *
         *
         *
         *
         */
void ext6(p)    short *p;
{
#ifdef  COMMENTS_NOT_CODE
        Let it be known that 'p' points to a table which is the
        right side of Table 36.
        The table is completley defined by functions ext1..ext6.
        The left side of the table is composed of entries which
        are digital representation (of the printer)
        on the defined grid.
        The order of the entries is such that they first vary
        along the first dimension, than along the second etc.
Example 1:      in this exmple the left side and the right side
        of the table contain the same values.

        int i,j,k;
        static short table[5*5*5*3]=
        {
/* order: cyan,magenta,yellow,  cyan,magenta,yellow, ...        */
/* at yellow = 0 */
/* at:           cyan=0,   cyan=64,   cyan=128,   cyan=192,  cyan=155 */
/*magenta=0 */  0,0,0,    0,0,64,    0,0,128,    0,0,192,    0,0,255,
/*      64 */   0,64,0,   0,64,64,   0,64,128,   0,64,192,   0,64,255,
/*     128 */   0,128,0,  0,128,64,  0,128,128,  0,128,192,  0,128,255,
/*     192 */   0,192,0,  0,192,64,  0,192,128,  0,192,192,  0,192,255,
/*     255 */   0,255,0,  0,255,64,  0,255,128,  0,255,192,  0,255,255,
/* at yellow = 64 */
/* at:
cyan=0,    cyan=64,   cyan=128,   cyan=192,   cyan=155    */
64,0,0,    64,0,64,   64,0,128,   64,0,192,   64,0,255,
64,64,0,   64,64,64,  64,64,128,  64,64,192,  64,64,255,
64,128,0,  64,128,64, 64,128,128, 64,128,192, 64,128,255,
64,192,0,  64,192,64, 64,192,128, 64,192,192, 64,192,255,
64,255,0,  64,255,64, 64,255,128, 64,255,192, 64,255,255,
/* at yellow = 128 */
/* at:
cyan=0,     cyan=64,    cyan=128,    cyan=192,    cyan=155    */
128,0,0,    128,0,64,   128,0,128,   128,0,192,   128,0,255,
128,64,0,   128,64,64,  128,64,128,  128,64,192,  128,64,255,
128,128,0,  128,128,64, 128,128,128, 128,128,_92, 128,128,255,
128,192,0,  128,192,64, 128,192,128, 128,192,192, 128,192,255,
128,255,0,  128,255,64, 128,255,128, 128,255,192, 128,255,255,
/*
 *
```

```
 *
 *
 *
 *
 *
 *
 *
 *
 * at yellow = 192 */
/* at:
cyan=0,      cyan=64,       cyan=128,       cyan=192,       cyan=155 */
192,0,0,     192,0,64,      192,0,128,      192,0,192,      192,0,255,
192,64,0,    192,64,64,     192,64,128,     192,64,192,     192,64,255,
192,128,0,   192,128,64,    192,128,128,    192,128,192,    192,128,255,
192,192,0,   192,192,64,    192,192,128,    192,192,192,    192,192,255,
192,255,0,   192,255,64,    192,255,128,    192,255,192,    192,255,255,
/* at yellow = 255 */
/* at:
cyan=0,      cyan=64,       cyan=128,       cyan=192,       cyan=155 */
255,0,0,     255,0,64,      255,0,128,      255,0,192,      255,0,255,
255,64,0,    255,64,64,     255,64,128,     255,64,192,     255,64,255,
255,128,0,   255,128,64,    255,128,128,    255,128,192,    255,128,255,
255,192,0,   255,192,64,    255,192,128,    255,192,192,    255,192,255,
255,255,0,   255,255,64,    255,255,128,    255,255,192,    255,255,255
        };
        ext1(3);              // 3 dimensions in input
        ext2(3);              // and 3 dimensions in output
        ext4(5);
        for(i=0;i<3;i++)      // all dimensions have 5 grid stops
          ext3(i,5);
        for(i=0;i<3;i++)
        {
          ext5(i,0,0);        // definition of the grid
          ext5(i,1,64);
          ext5(i,2,128);
          ext5(i,3,192);
          ext5(i,4,255);
        }
        ext6(table);
#endif
}
int ext8()
{
#ifdef  COMMENTS_NOT_CODE
        Supply the dimension of the digital representation
        which is the output of the resulting Table 36.
        Example 1: 3 for cyan,magenta,yellow
        Example 2: 4 for cyan,magenta,yellow,black
#endif
/*Example:*/    return(3);
};
/*
 *
```

```
 *
 *
 *
 *
 *
 *
 *
 */
int ext9()
{
#ifdef  COMMENTS_NOT_CODE
        Supply the dimension of the digital representation
        which is the input of the resulting Table 36.
        Example 1: 3 for cyan,magenta,yellow
        Example 2: 4 for cyan,magenta,yellow,black
#endif
/*Example:*/    return(3);
};
int  ext10(i,j)          int i,j;
{
#ifdef  COMMENTS_NOT_CODE
        Supply the number at the 'j'-th grid stop at
        the 'i'-th input dimension.
Example 1: if the cyan grid is
        (0,10,20,30,40,50,60,70,80,90,100)
        than k=90 for i=0 and j=9.
#endif
/*Example 2:*/
        int k = i*64;
        if(k==256)      k=255;
        return(k);
}
int  ext11()
{
#ifdef  COMMENTS_NOT_CODE
        Supply the RGB dimension of tables 1 and 2 in Fig. 4.
#endif
/*Example:*/    return(3);
}
int  ext12()
{
#ifdef  COMMENTS_NOT_CODE
        Supply the number of entries in a CMY vs. RGB table
        such as 1 and 2.
#endif
/*Example:*/    return(512);
}
/*
 *
```

```
 *
 *
 *
 *
 *
 *
 *
 */
int  ext13(i,j)        int i,j;
{
#ifdef  COMMENTS_NOT_CODE
        Supply value of entry 'i', separation 'j' of the
        CMY side of a RGB|CMY table such as 1 and 2 in Fig 4.
#endif
/*Dummy:*/      return(0);
}
int  ext14(i,j)        int i,j;
{
#ifdef  COMMENTS_NOT_CODE
        Supply value of entry 'i', separation 'j' of the
        RGB side of a RGB|CMY table such as 1 and 2 in Fig 4.
#endif
/*Dummy:*/      return(0);
}
int  ext15()
{
#ifdef  COMMENTS_NOT_CODE
        Supply the number of entries in a RGB table such as
        1 or 2 in Fig 4., but it which the CMY values are
        not listed but can be caculated from
        a description of a grid such as described above.
#endif
/*Example:*/    return(512);
}
int  ext16(p,i,j)      short *p; int i,j;
{
#ifdef  COMMENTS_NOT_CODE
        Supply an item of a table 'p', such as described in ext15,
        in the 'i'-th entry, in the 'j'-th separation.
#endif
/*Dummy:*/      return(0);
}
void ext17(p)   char *p;
{
#ifdef  COMMENTS_NOT_CODE
Sometimes in the description above it is not clear if an ext..()
function reffers to Table 36 of the printer, or to Table 32 of
the proofer.
'p' is a string of characters terminated by null.
If the last character, before the null, is 'a'
than the following ext..() calls should reffer to the proofer.
If the last character, before the null, is 'b'
than the following ext..() calls should reffer to the printer.
#endif
}
/*
 *
```

```
 *                                                          Page 7
 *
 *
 *
 *
 *
 *
 *
 **************************** Inerface to PAT2 *****************
#ifdef  COMMENTS_NOT_CODE
The maximum number of dimensions in color space here is 4.
#endif

int     ext20(i)        int i;
{
#ifdef  COMMENTS_NOT_CODE
This is the same as ext8(), but it is used as input for pat2.
'i' is in the range 0 to 3
A value of 1 for 'i' = 3 signals the existance
of only 3 separations.
The returned value must be in the range 0 to 10
#endif
}
int     ext21(i,j)      int i,j;
{
#ifdef  COMMENTS_NOT_CODE
        This is the same as ext10(), but as input for pat2.
        'i' is in the range 0 to 3
        'j' is in the range 0 to 10
#endif
}
short   *ext22()
{
#ifdef  COMMENTS_NOT_CODE
        This supplies pat2 with Table 36,
        and in the same format as in ext6().
#endif
}
int     ext23(i)        int i;
{
#ifdef  COMMENTS_NOT_CODE
This defines the dimensions of Table 42 in the same way
as ext8() and ext20(), but reffering to the output of pat2.
'i' is in the range 0 to 3
#endif
}
/*
 *
```

```
 *                                                  Page 8
 *
 *
 *
 *
 *
 *
 */
int     ext24(i,j)      int i,j;
{
#ifdef  COMMENTS_NOT_CODE
This defines the dimensions of Table 42 in the same way
as ext10() and ext21(), but reffering to the output of pat2.
'i' is in the range 0 to 3
'j' is in the range 0 to 32
#endif
}
void    ext27(iv,i,f)   int iv[4],i;    float f;
{
#ifdef  COMMENTS_NOT_CODE
This routine gets the output from pat2..
'iv' is a vector of values in the range 0..32, indexes
to the grid defined using ext24, and so defining a point in the
color space.  This is an entry in CMY to CMY Table 42.
'f' is the 'i'-th value  written in this table, at this entry.
pat2 calls ext27 for all entries 'iv' and all dimensions 'i'.
#endif
}
```

Claims

1. A technique for calibrating a color processing device comprising the steps of:

   comparing a first digital representation of a colored image with a second digital representation thereof. said first digital representation defining a plurality of first non-scalar color values, said second digital representation defining a plurality of second non-scalar color values corresponding to the plurality of said first non-scalar color values, thereby to provide a transformation pairing each individual one of said first non-scalar color values with a value relatively close to the corresponding one of said second non-scalar color values: and

   employing at least the transformation to control operation of said color processing device to be calibrated.

   whereby the color processing device may be calibrated generally without reference to human aesthetic judgement.

2. Apparatus for sampling the color processing characteristics of a color processing device, said color processing device being operative to convert a first representation of a colored image to a second representation thereof, said sampling apparatus comprising:

   a first representation of a colored image characterized in that processing said first representation of said colored image with the color processing device provides a second representation of said colored image which defines a provided plurality of color values, each individual one of said provided plurality of color values being substantially equal to a corresponding one of a predetermined plurality of color values falling within the range of said color processing device.

3. A method of constructing apparatus for sampling the color processing characteristics of a color processing device, said color processing device being operative to convert a first representation of a colored image to a second representation thereof,

   the method comprising the step of repeating at least once the steps of:

   providing first and second representations of a colored image, said representations respectively comprising a first multiplicity of first color values and a second multiplicity of second color values corresponding thereto, said first and second representations being characterized in that processing said

first representation with said color processing device defines said second representation.

comparing the first representation of the colored image with the second representation thereof, thereby to provide a transformation, characterized in that operating said transformation on each individual one of said second multiplicity of second color values gives a value substantially equal to the corresponding one of said first multiplicity of first color values: and

operating the transformation on said first representation of the colored image, thereby to provide a third representation thereof.

4. A technique for quality control of a color processing device operative to convert a first representation of a colored image to a second representation thereof, the technique comprising the steps of:

providing apparatus for sampling the color processing characteristics of the color processing device, said sampling apparatus comprising a first representation of a colored image characterized in that processing said first representation of said colored image with the color processing device provides a second representation of said colored image which defines a provided plurality of color values, each individual one of said provided plurality of color values being substantially equal to a corresponding one of a predetermined plurality of color values falling within the range of said color processing device;

processing said sampling apparatus on said color processing device: and

employing the results of the processing step of control oepration of the color processing device.

5. A technique for repeatibility testing of a color processing device operative to convert a first representation of a colored image to a second representation thereof, the technique comprising the steps of:

providing apparatus for sampling the color processing characteristics of the color processing device, said sampling apparatus comprising a first representation of a colored image characterized in that processing said first representation of said colored image with the color processing device provides a second representation of said colored image which defines a provided plurality of color values, each individual one of said provided plurality of color values being substantially equal to a corresponding one of a predetermined plurality of color values falling within the range of said color processing device;

processing said sampling apparatus on said color processing device;

repeating said step of processing on at least one further occasion; and

comparing the results of at least two repetitions of said processing steps.

6. A method for transforming an element of a domain of a first color printing device to an element of a domain of a second color printing device, the method comprising the steps of:

providing a first transformation from a first digital representation of a colored image to a second digital representation thereof and a second transformation from a third digital representation of a colored image to a fourth digital representation thereof, said second transformation corresponding to said second color printing device, said first transformation corresponding to said first color printing device and said second and fourth digital representations being defined within a single color space; and

comparing said first transformation with said second transformation.

7. A method for quantifying the appearance of an analog representation of a location of a colored image comprising the step of:

providing an n-dimensional representation of said location, wherein n is at least 4.

8. Apparatus for transforming an element of a domain of a first color printing device to an element of a domain of a second color printing device comprising;

means for providing a first transformation from a first digital representation of a colored image to a second digital representation thereof and a second transformation from a third digital representation of a colored image to a fourth digital representation thereof, said second transformation corresponding to said second color printing device, said first transformation corresponding to said first color printing device and said second and fourth digital representations being defined within a single color space; and

means for comparing said first transformation with said second transformation.

9. Apparatus for quantifying the appearance of an analog representation of a location of a colored image comprising;

means for providing an n-dimensional representation of said location, wherein n is at least 4.

10. A system for converting between a digital and an analog representation of an image comprising:
a translation system for translating a reading and writing point across a substrate.
a color proofer comprising said writing point for writing. from said digital representation, said analog representation onto said substrate; and
a color reading system comprising said reading point for reading said digital representation from said analog representation.

11. A technique for controlling the operation of an image processing device having a first color coordinate system and comprising the steps of:
receiving a digital representation of a color image defined in a second color coordinate system;
providing a transformation between said first color coordinate system and said second color coordinate system;
employing said transformation to transform said digital representation of said color image into a transformed digital representation of said color image in said first color coordinate system; and
employing said image processing device to modify said transformed digital representation of said color image.

12. An image processing device having a first color coordinate system comprising:
means for receiving a digital representation of a color image defined in a second color coordinate system;
transformation construction means for providing a transformation between said first color coordinate system and said second color coordinate system: and
means for employing said transformation to transform said digital representation of said color image into a transformed digital representation of said color image in said first color coordinate system,
wherein said image processing device is operative to modify said transformed digital representation of said color image.

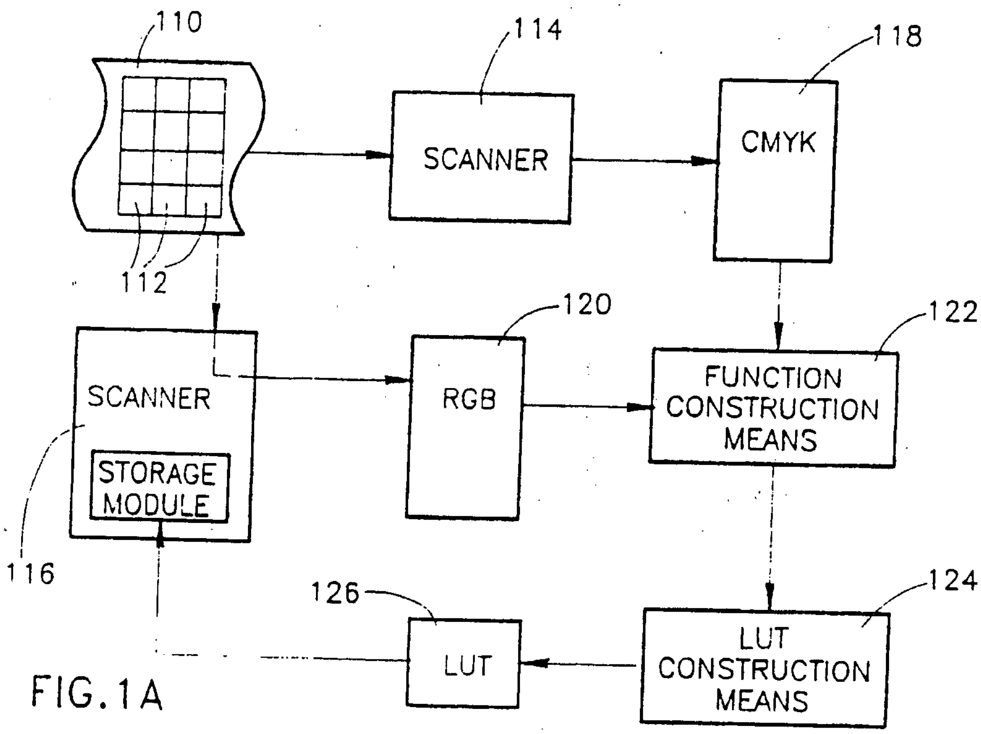

110
114
118
SCANNER
CMYK
112
120
122
SCANNER
RGB
FUNCTION CONSTRUCTION MEANS
STORAGE MODULE
116
126
124
LUT
LUT CONSTRUCTION MEANS

FIG.1A

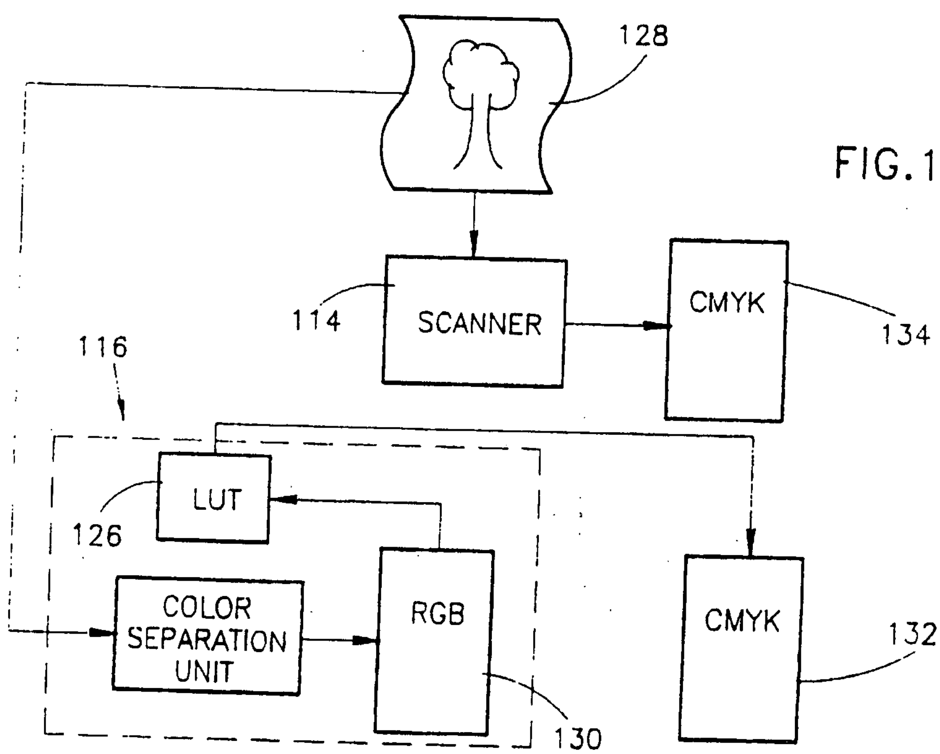

128
114
SCANNER
CMYK
134
116
LUT
126
COLOR SEPARATION UNIT
RGB
CMYK
132
130

FIG.1B

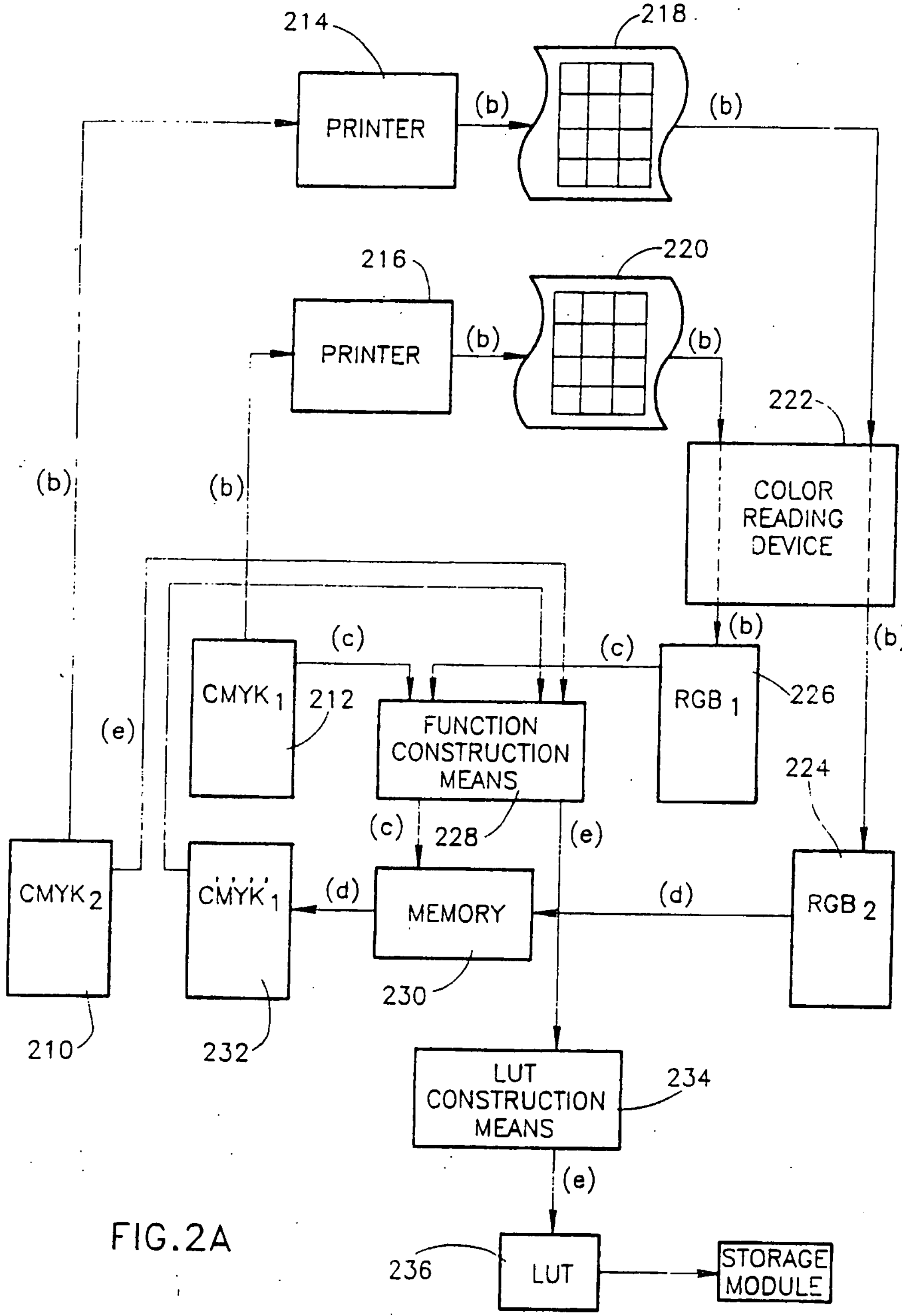

214
218
PRINTER
(b)
(b)
216
220
PRINTER
(b)
(b)
222
COLOR
READING
DEVICE
(b)
(b)
(b)
(b)
(c)
(c)
CMYK 1
212
FUNCTION
CONSTRUCTION
MEANS
RGB 1
226
(e)
224
(c)
228
(e)
CMYK 2
CMYK 1
(d)
MEMORY
(d)
RGB 2
230
210
232
LUT
CONSTRUCTION
MEANS
234
(e)
236
LUT
STORAGE
MODULE

FIG.2A

FIG.2B

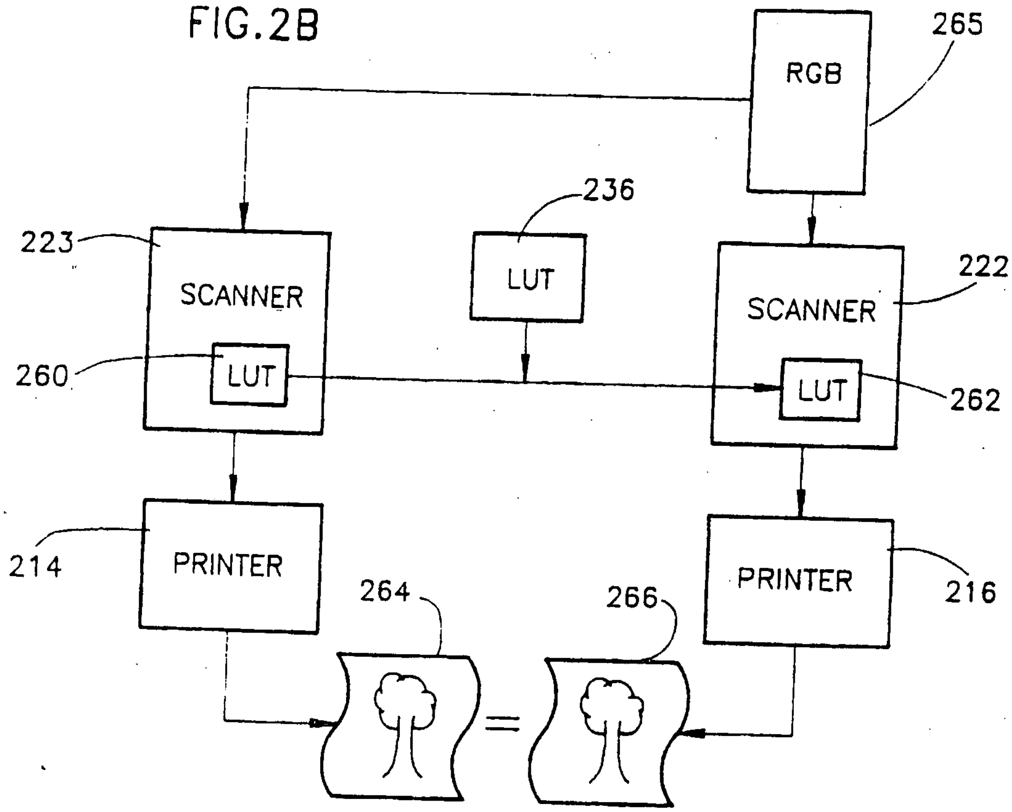
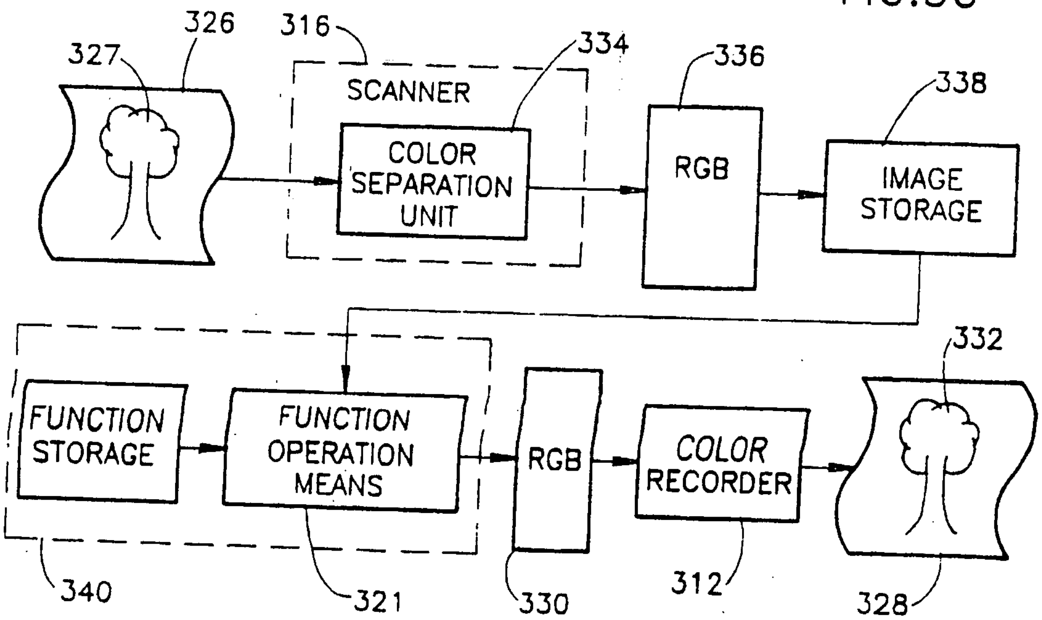

RGB
265
236
LUT
223
SCANNER
SCANNER
222
260
LUT
LUT
262
214
PRINTER
264
266
PRINTER
216

FIG.3C


326
316
334
336
338
327
SCANNER
COLOR SEPARATION UNIT
RGB
IMAGE STORAGE
332
FUNCTION STORAGE
FUNCTION OPERATION MEANS
RGB
COLOR RECORDER
340
321
330
312
328

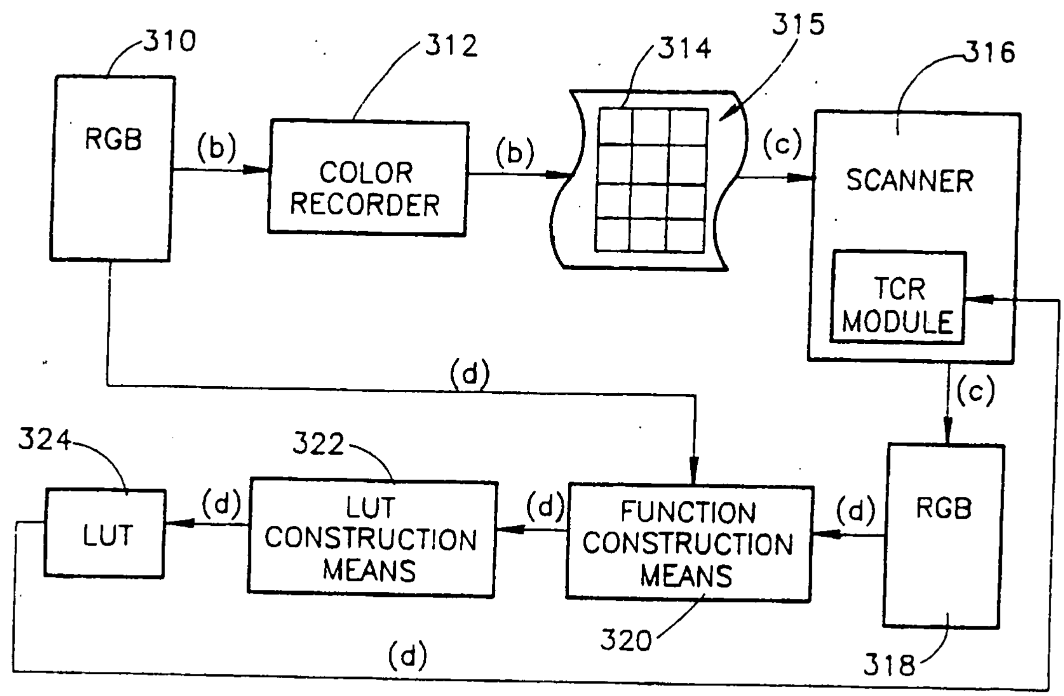

310
RGB
(b)
312
COLOR RECORDER
(b)
314
315
(c)
316
SCANNER
TCR MODULE
(c)
(d)
324
LUT
(d)
322
LUT CONSTRUCTION MEANS
(d)
FUNCTION CONSTRUCTION MEANS
(d)
RGB
320
318
(d)

FIG.3A

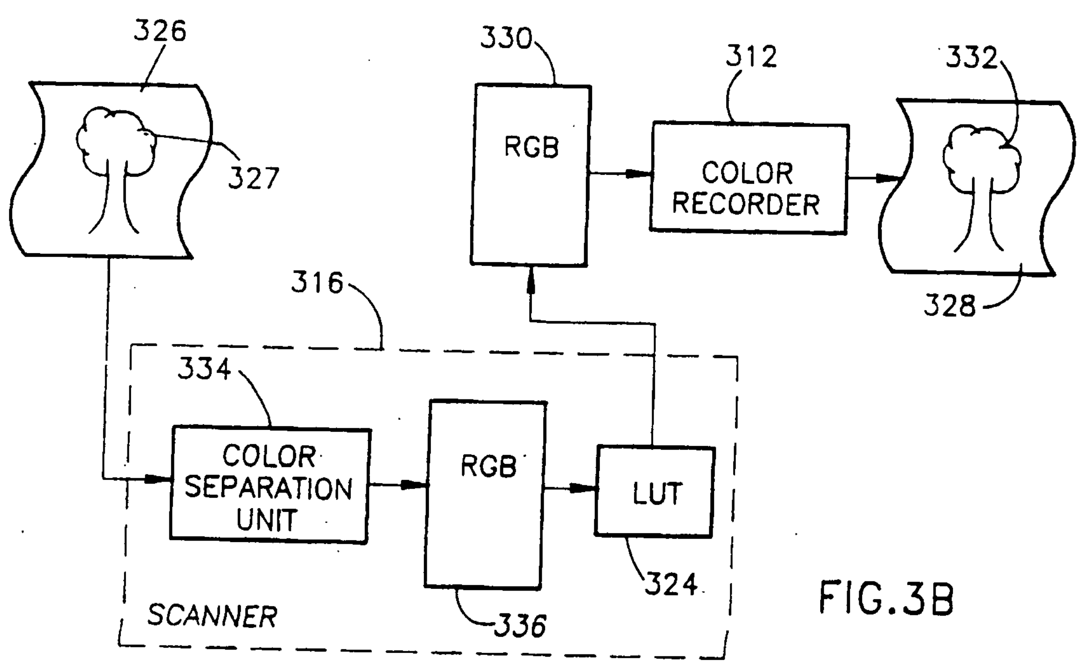

326
327
330
RGB
312
COLOR RECORDER
332
328
316
334
COLOR SEPARATION UNIT
RGB
LUT
324
SCANNER
336

FIG.3B

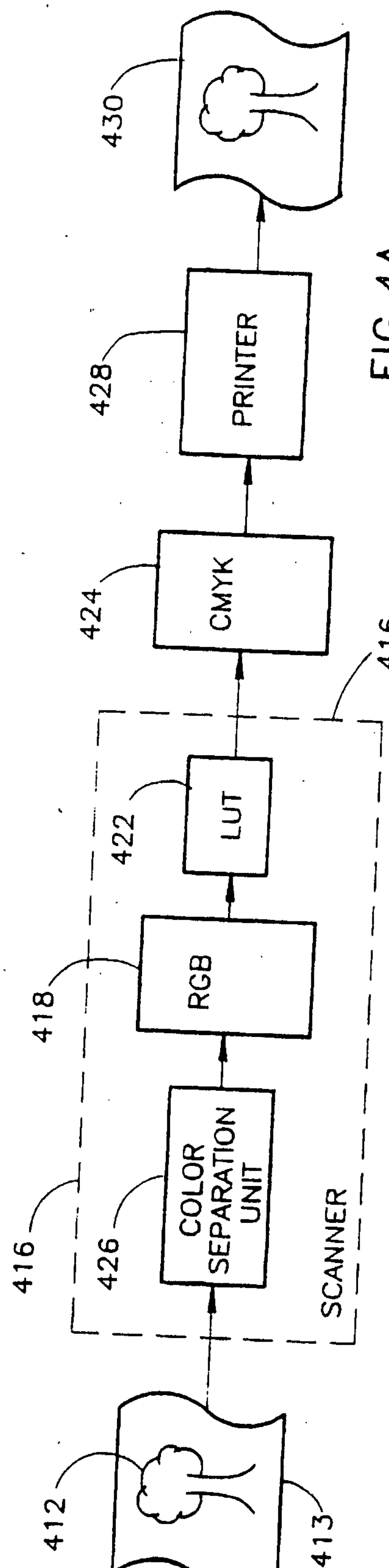

412
413
416
426
COLOR SEPARATION UNIT
SCANNER
418
RGB
422
LUT
416
424
CMYK
428
PRINTER
430

FIG.4A

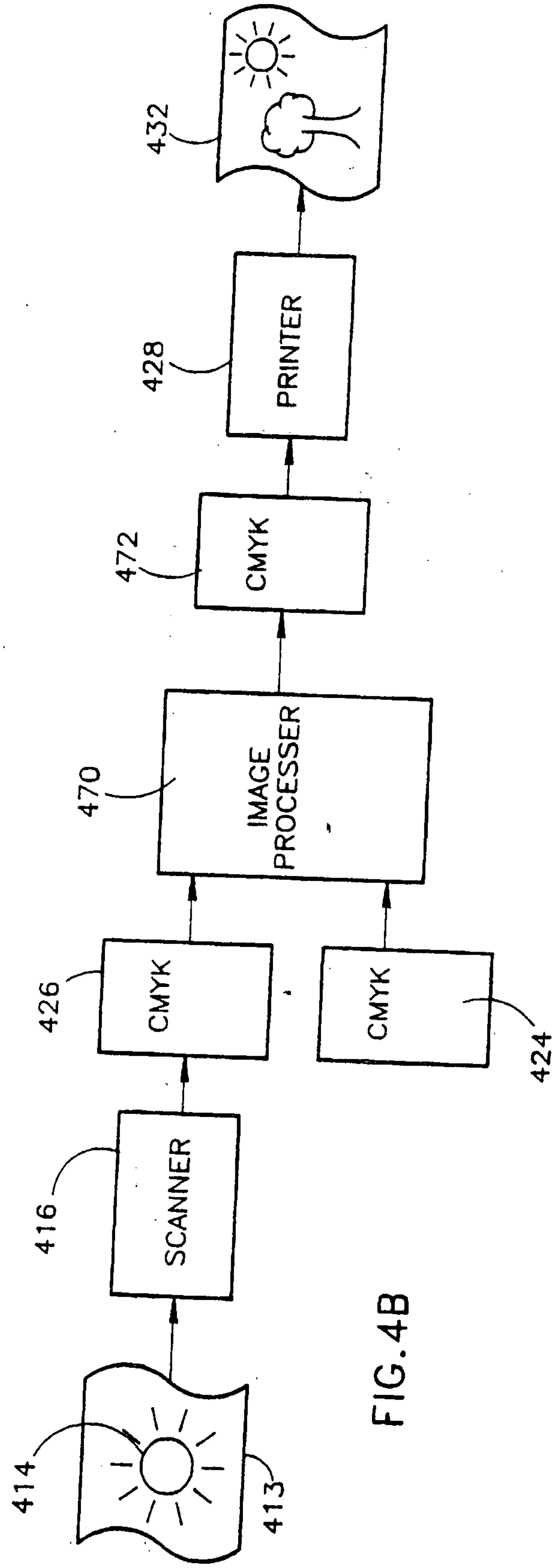

414
413
416
SCANNER
426
CMYK
CMYK
424
470
IMAGE PROCESSER
472
CMYK
428
PRINTER
432

FIG.4B

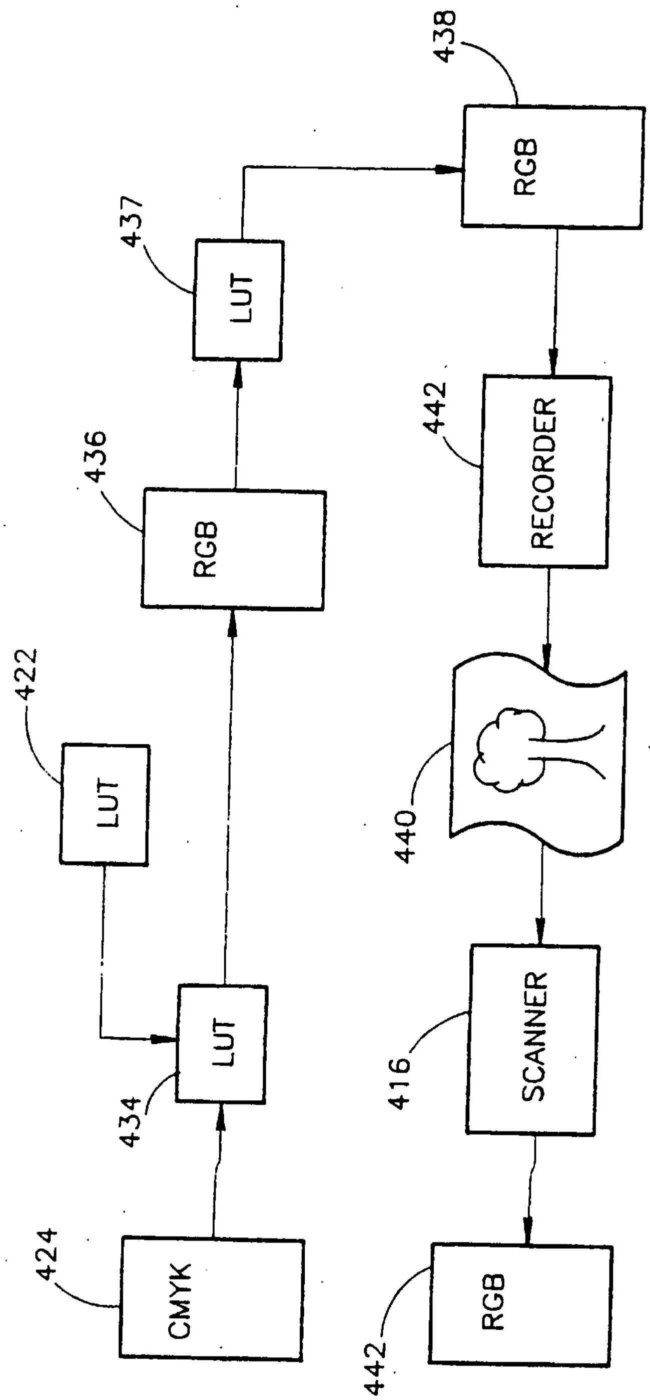

424
CMYK
434
LUT
422
LUT
436
RGB
437
LUT
438
RGB
442
RECORDER
440
416
SCANNER
442
RGB

FIG. 4C

FIG.4D

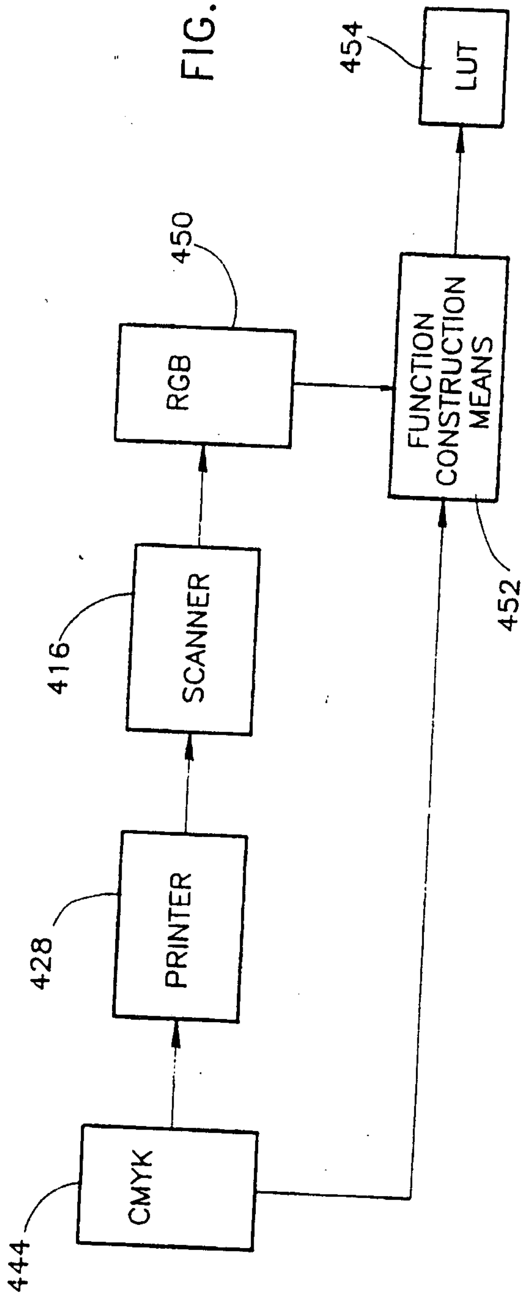
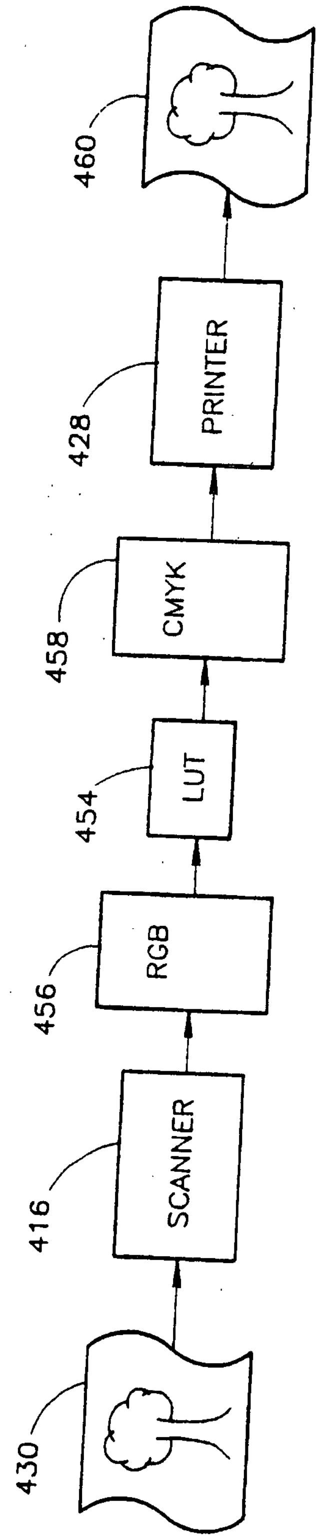

444
CMYK
428
PRINTER
416
SCANNER
450
RGB
FUNCTION CONSTRUCTION MEANS
452
454
LUT

FIG.4E


430
416
SCANNER
456
RGB
454
LUT
458
CMYK
428
PRINTER
460

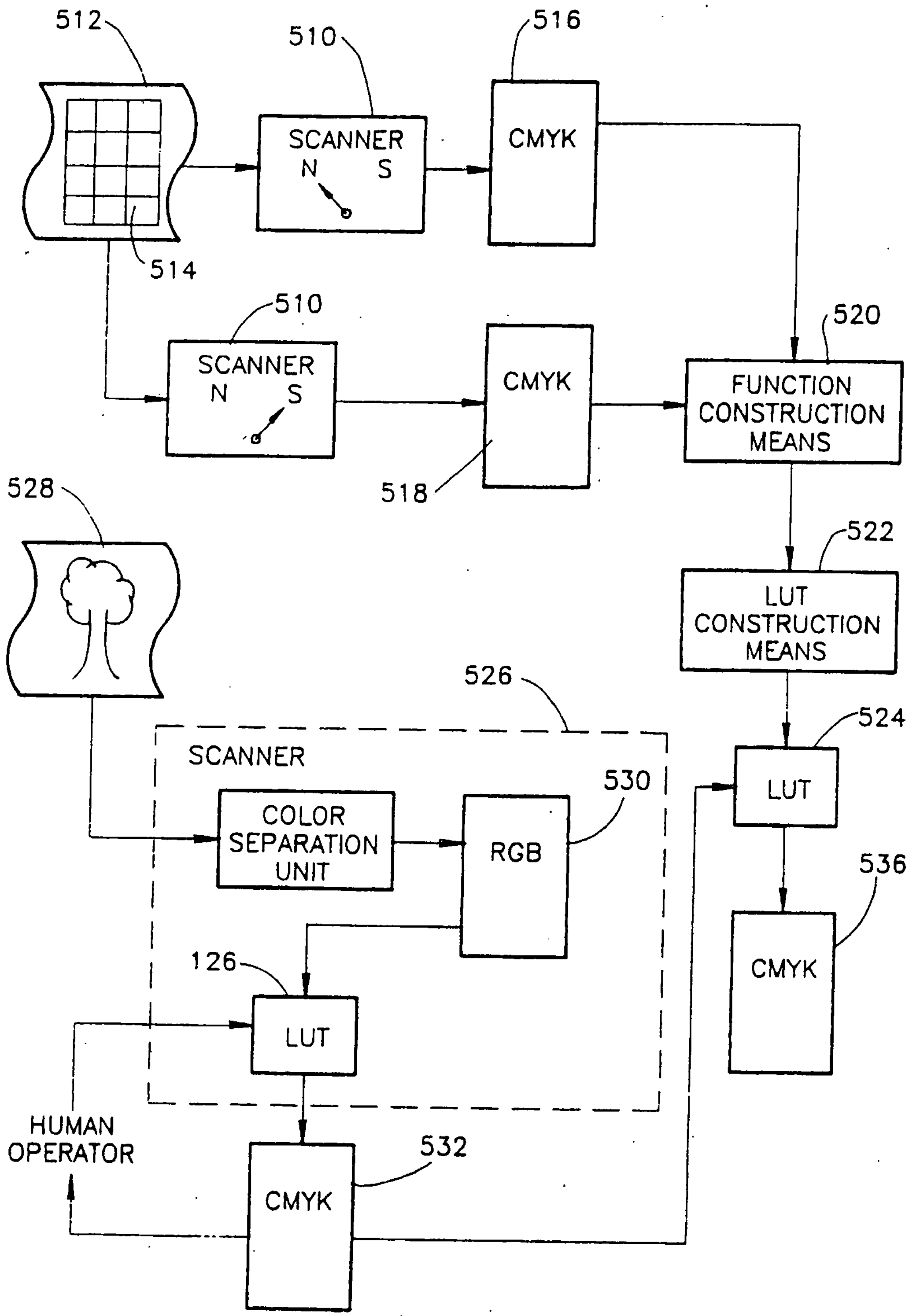

512
510
516
SCANNER
N S
CMYK
514
510
520
SCANNER
N S
CMYK
FUNCTION CONSTRUCTION MEANS
518
528
522
LUT CONSTRUCTION MEANS
526
524
SCANNER
530
LUT
COLOR SEPARATION UNIT
RGB
536
126
CMYK
LUT
HUMAN OPERATOR
532
CMYK

FIG.5A

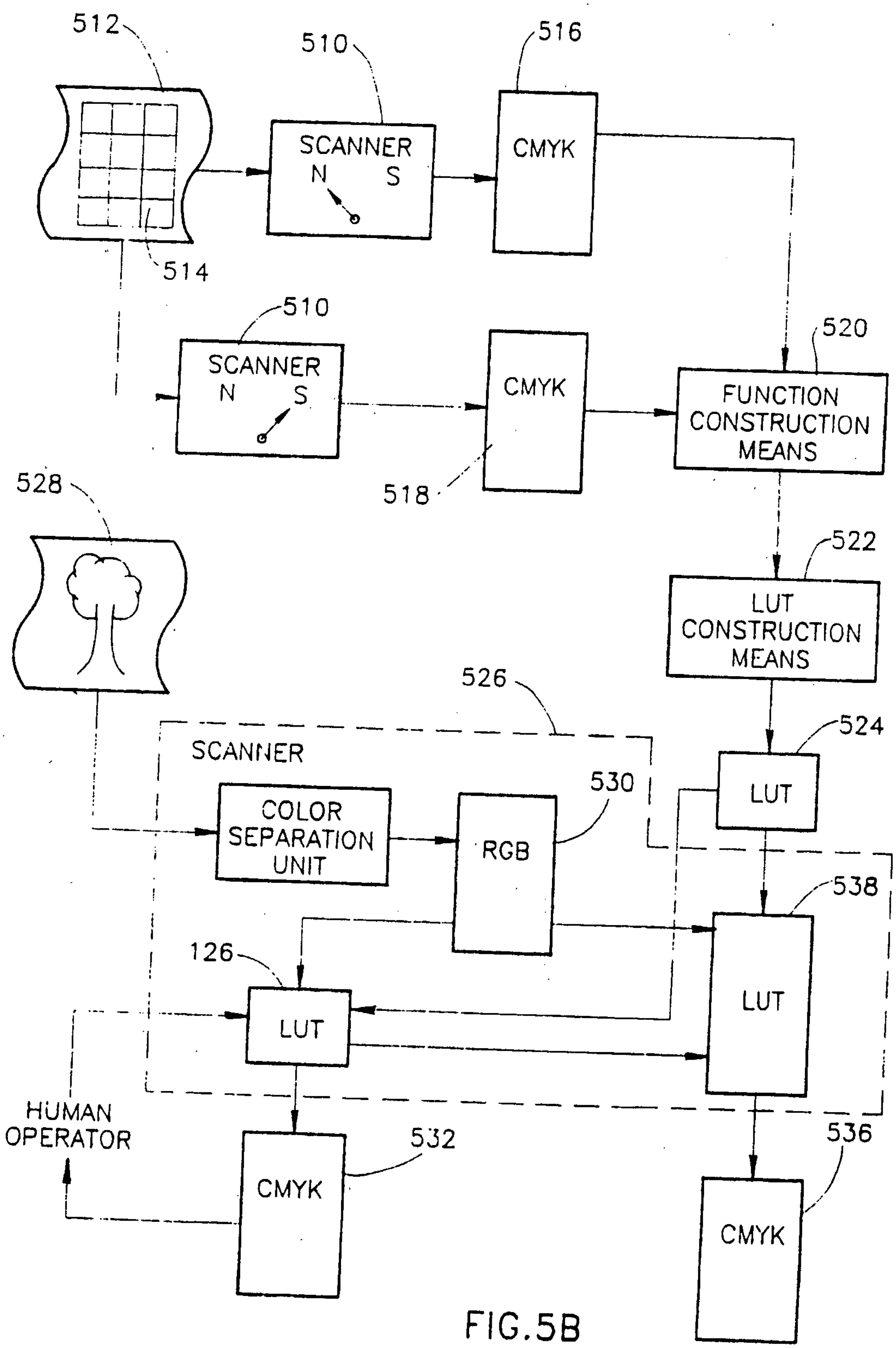

512
510
516
SCANNER
N S
CMYK
514
510
520
SCANNER
N S
CMYK
FUNCTION CONSTRUCTION MEANS
518
528
522
LUT CONSTRUCTION MEANS
526
524
SCANNER
530
LUT
COLOR SEPARATION UNIT
RGB
538
126
LUT
LUT
HUMAN OPERATOR
532
536
CMYK
CMYK

FIG.5B

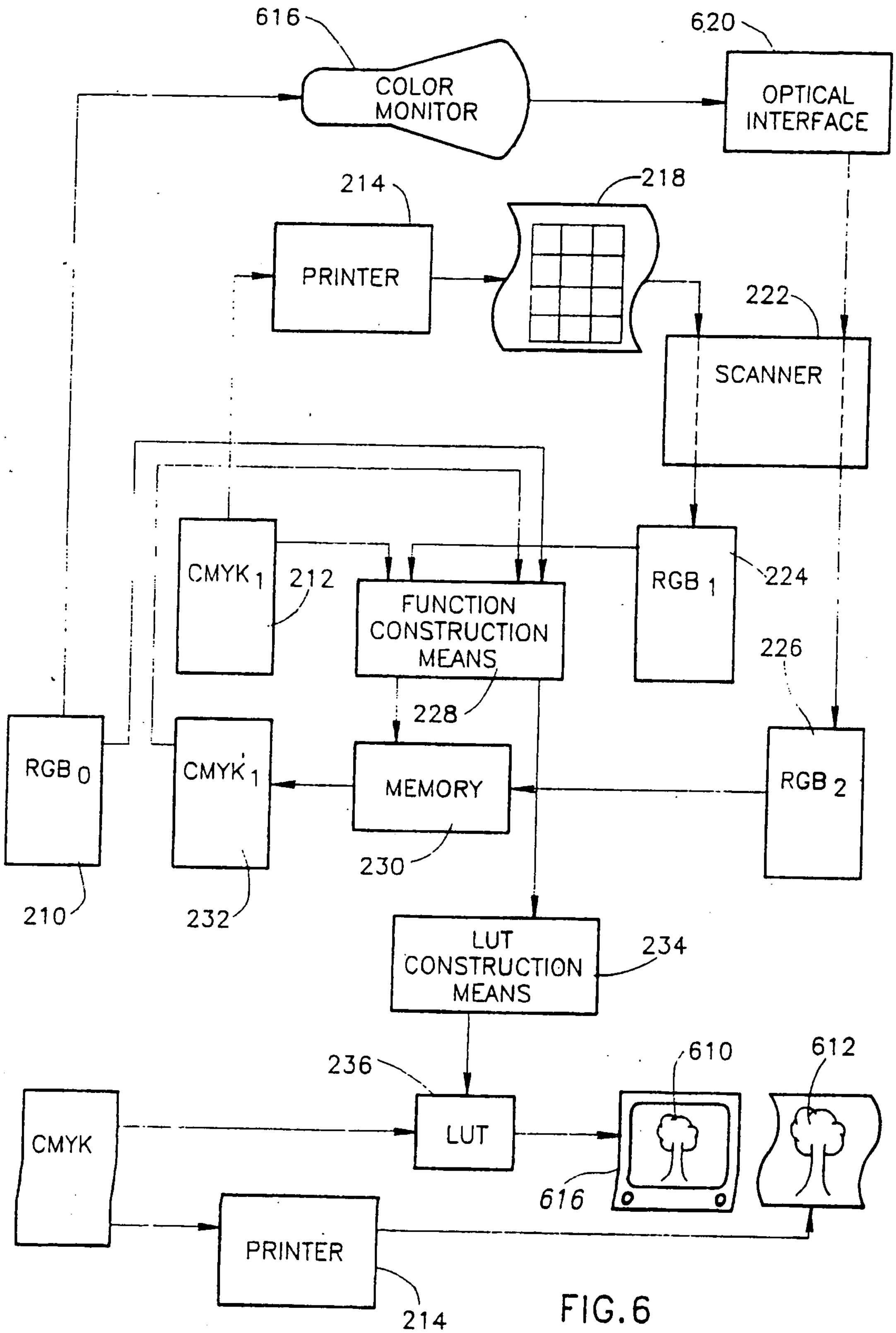

616
COLOR MONITOR
620
OPTICAL INTERFACE
214
PRINTER
218
222
SCANNER
CMYK 1
212
FUNCTION CONSTRUCTION MEANS
RGB 1
224
226
228
RGB 0
CMYK 1
MEMORY
RGB 2
230
210
232
LUT CONSTRUCTION MEANS
234
236
610
612
CMYK
LUT
616
PRINTER
214

FIG.6

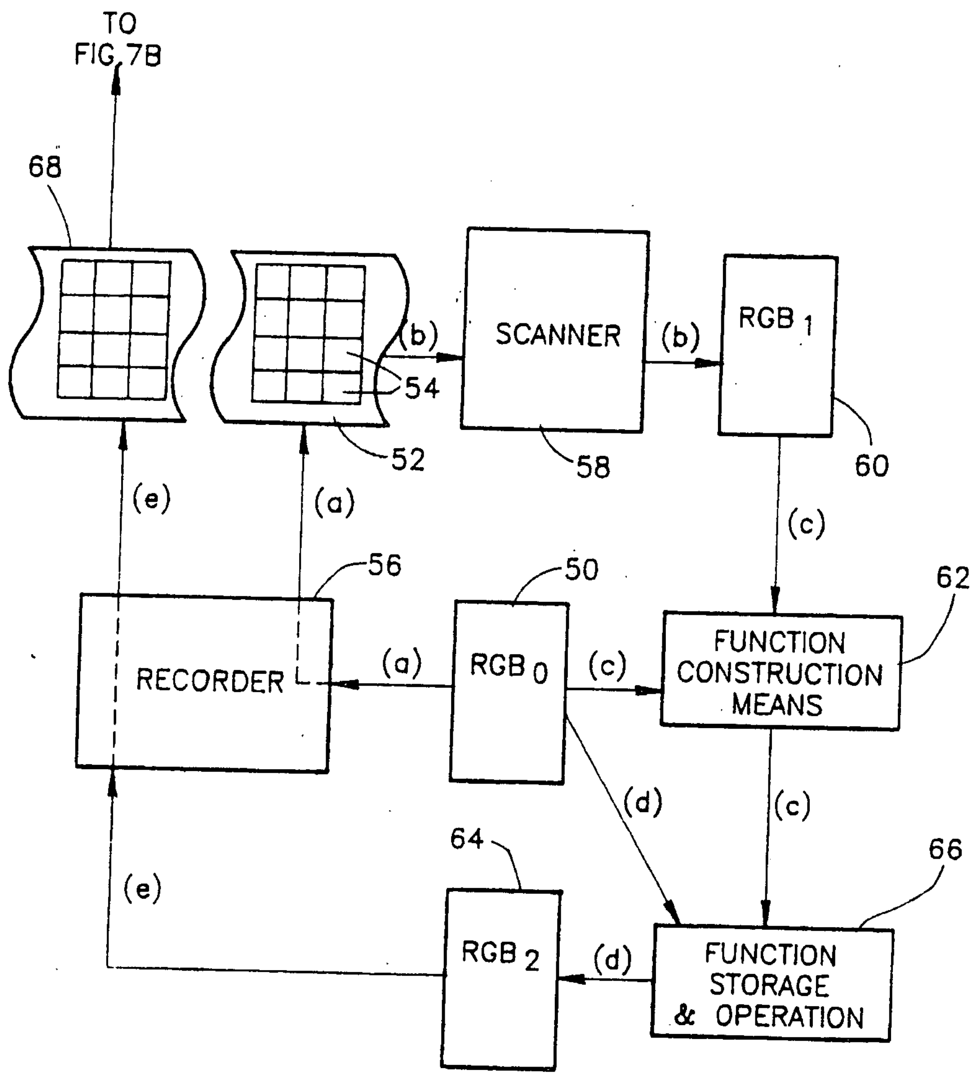

TO
FIG.7B
68
(b)
SCANNER
(b)
RGB 1
54
52
58
60
(e)
(a)
(c)
56
50
62
RECORDER
(a)
RGB 0
(c)
FUNCTION
CONSTRUCTION
MEANS
(d)
(c)
64
66
(e)
RGB 2
(d)
FUNCTION
STORAGE
& OPERATION

FIG.7A

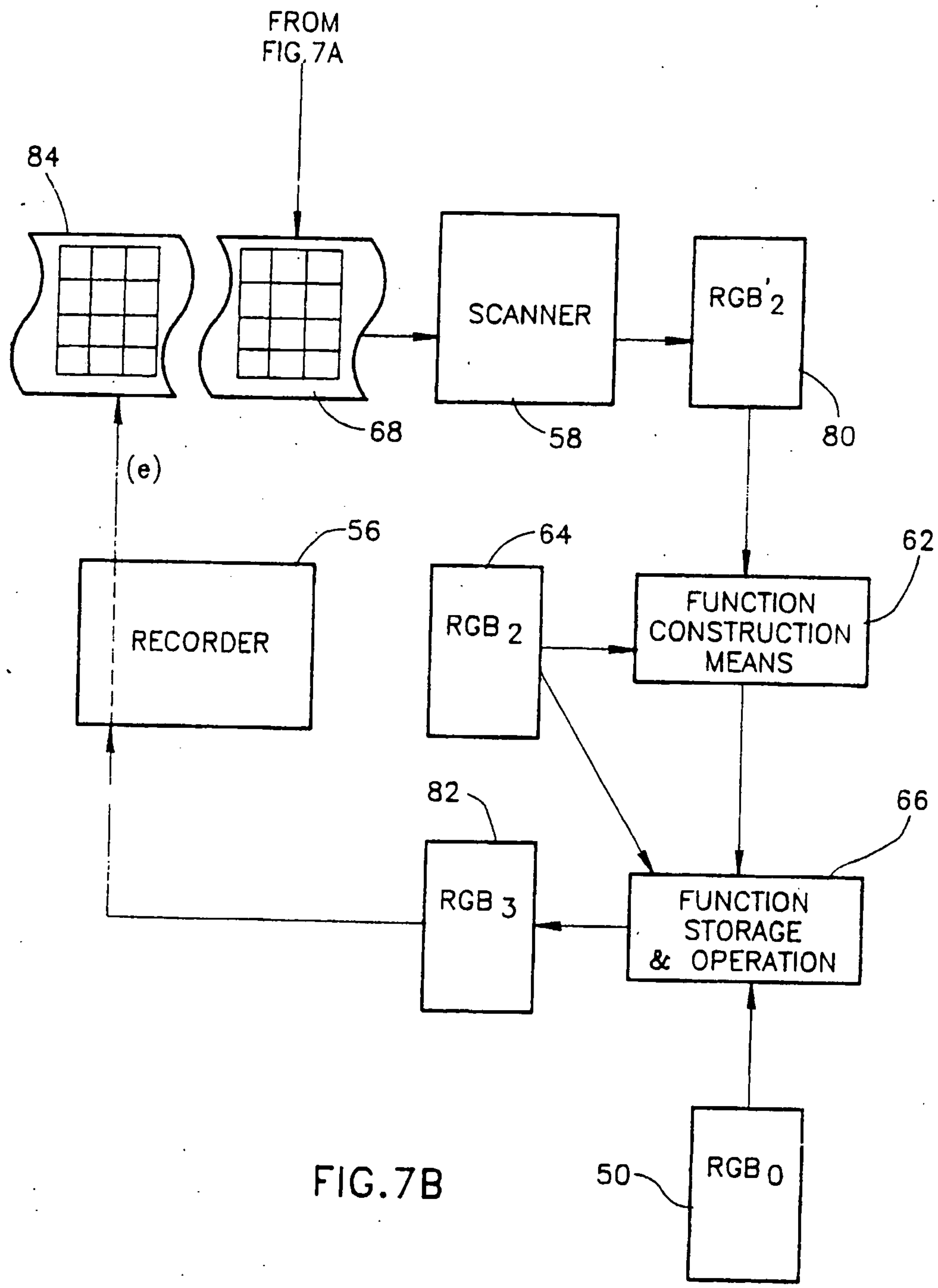

FROM
FIG. 7A
84
68
SCANNER
58
RGB'2
80
(e)
56
RECORDER
64
RGB 2
FUNCTION
CONSTRUCTION
MEANS
62
82
66
RGB 3
FUNCTION
STORAGE
& OPERATION
50
RGB 0

FIG.7B

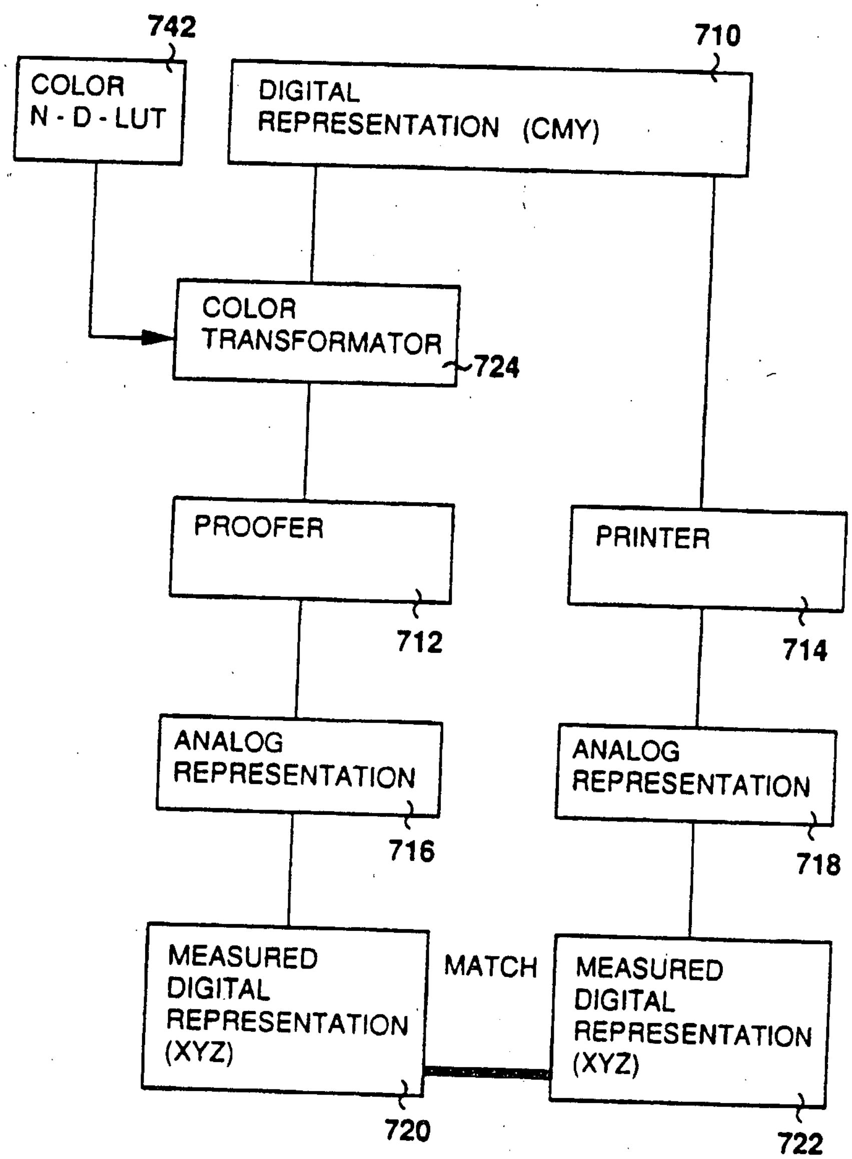

742
COLOR
N - D - LUT
710
DIGITAL
REPRESENTATION (CMY)
COLOR
TRANSFORMATOR
724
PROOFER
712
PRINTER
714
ANALOG
REPRESENTATION
716
ANALOG
REPRESENTATION
718
MEASURED
DIGITAL
REPRESENTATION
(XYZ)
720
MATCH
MEASURED
DIGITAL
REPRESENTATION
(XYZ)
722

FIG. 8

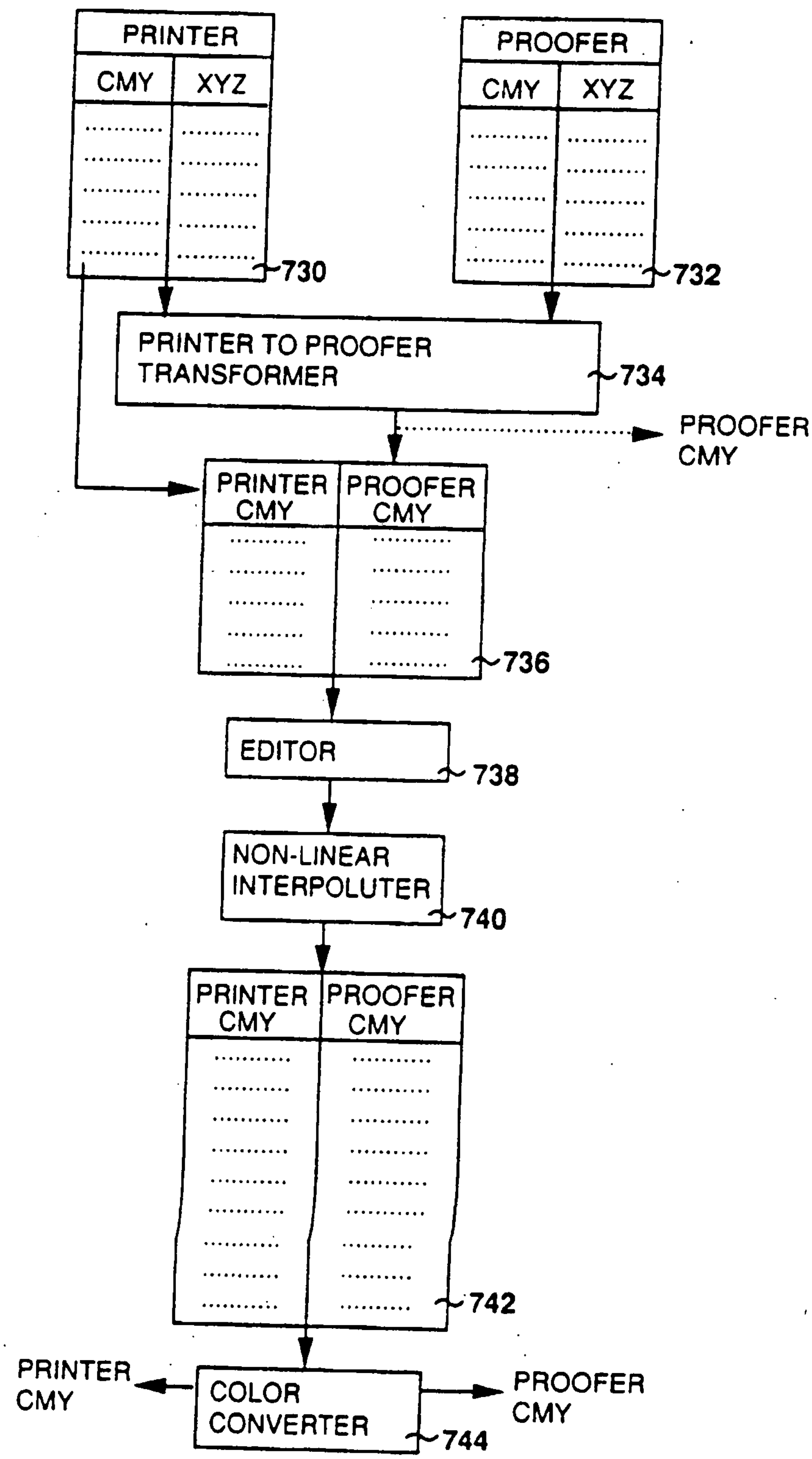

PRINTER
CMY
XYZ
730
PROOFER
CMY
XYZ
732
PRINTER TO PROOFER TRANSFORMER
734
PROOFER CMY
PRINTER CMY
PROOFER CMY
736
EDITOR
738
NON-LINEAR INTERPOLUTER
740
PRINTER CMY
PROOFER CMY
742
PRINTER CMY
COLOR CONVERTER
744
PROOFER CMY

FIG. 9

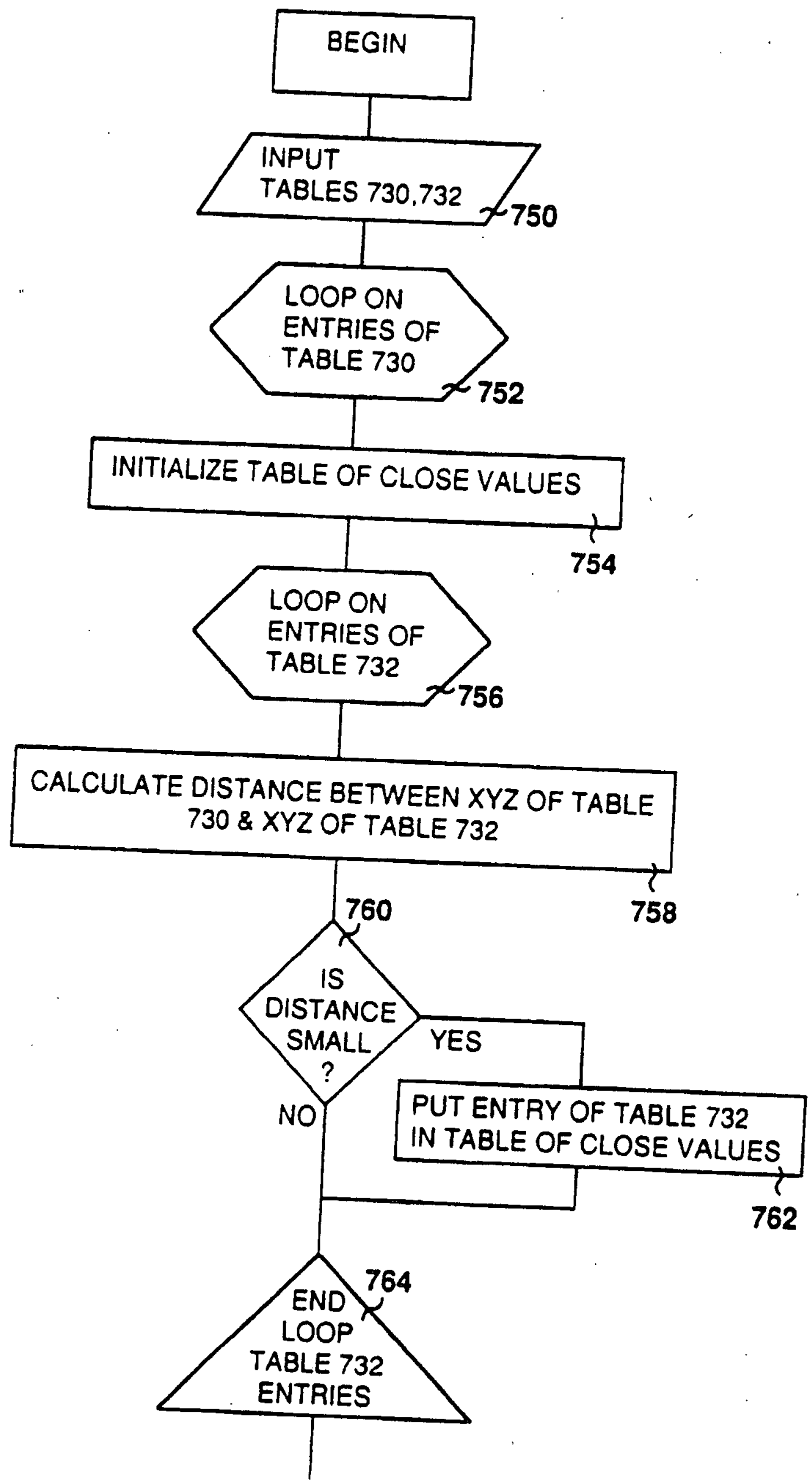

BEGIN
INPUT TABLES 730,732
750
LOOP ON ENTRIES OF TABLE 730
752
INITIALIZE TABLE OF CLOSE VALUES
754
LOOP ON ENTRIES OF TABLE 732
756
CALCULATE DISTANCE BETWEEN XYZ OF TABLE 730 & XYZ OF TABLE 732
758
760
IS DISTANCE SMALL ?
YES
NO
PUT ENTRY OF TABLE 732 IN TABLE OF CLOSE VALUES
762
764
END LOOP TABLE 732 ENTRIES

FIG.10/1

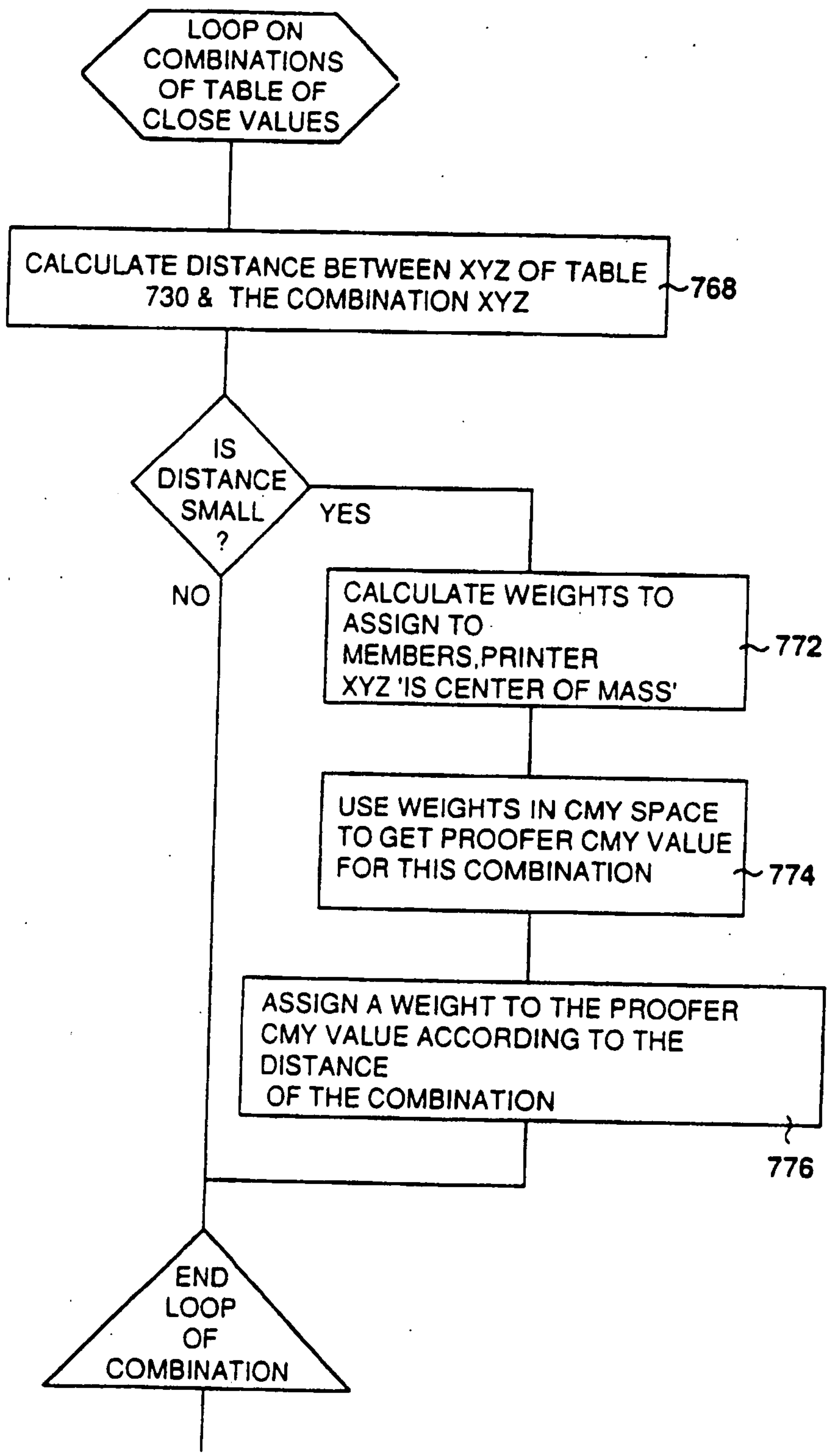

LOOP ON COMBINATIONS OF TABLE OF CLOSE VALUES
CALCULATE DISTANCE BETWEEN XYZ OF TABLE 730 & THE COMBINATION XYZ
768
IS DISTANCE SMALL ?
YES
NO
CALCULATE WEIGHTS TO ASSIGN TO MEMBERS,PRINTER XYZ 'IS CENTER OF MASS'
772
USE WEIGHTS IN CMY SPACE TO GET PROOFER CMY VALUE FOR THIS COMBINATION
774
ASSIGN A WEIGHT TO THE PROOFER CMY VALUE ACCORDING TO THE DISTANCE OF THE COMBINATION
776
END LOOP OF COMBINATION

FIG.10/2

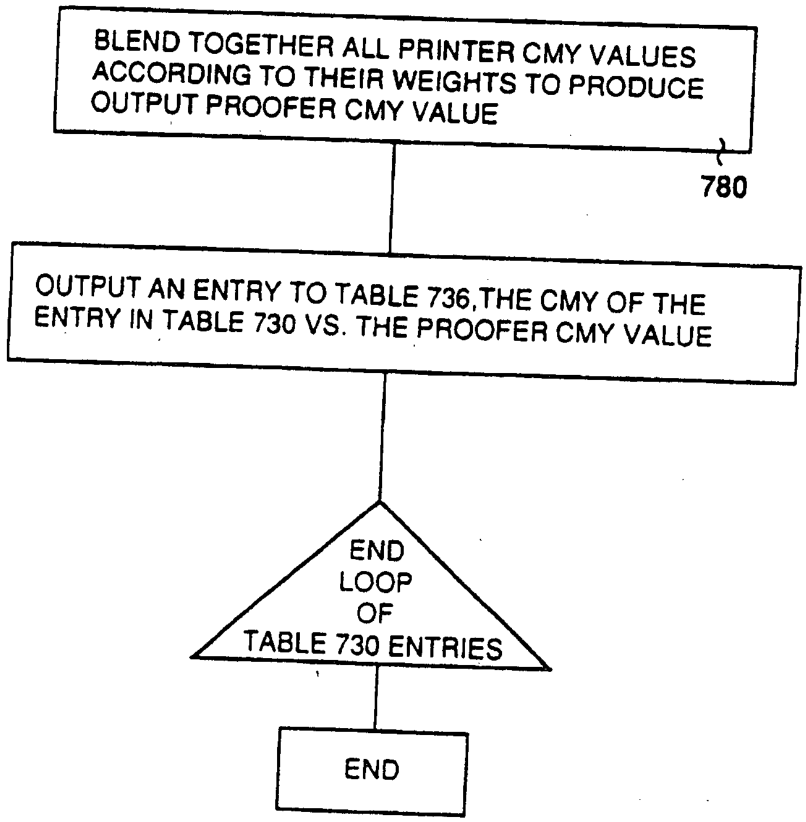

BLEND TOGETHER ALL PRINTER CMY VALUES ACCORDING TO THEIR WEIGHTS TO PRODUCE OUTPUT PROOFER CMY VALUE
780
OUTPUT AN ENTRY TO TABLE 736,THE CMY OF THE ENTRY IN TABLE 730 VS. THE PROOFER CMY VALUE
END LOOP OF TABLE 730 ENTRIES
END

FIG.10/3

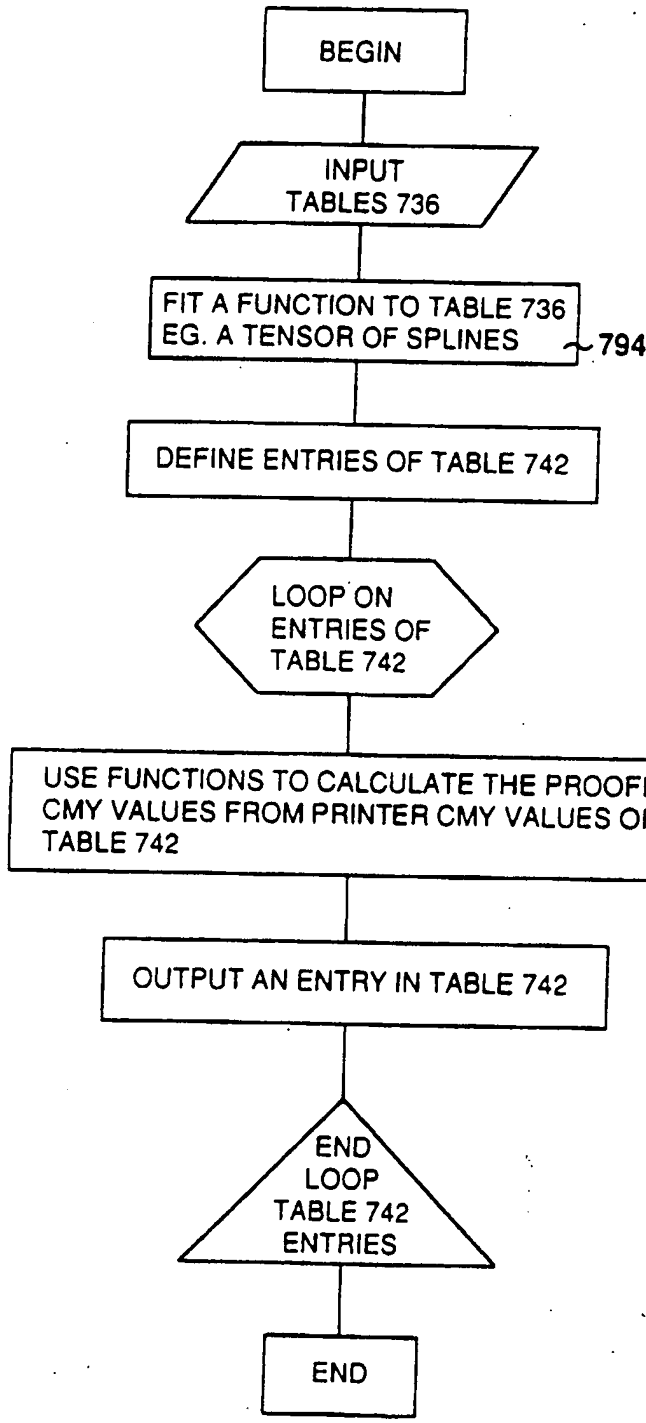

BEGIN
INPUT TABLES 736
FIT A FUNCTION TO TABLE 736 EG. A TENSOR OF SPLINES
794
DEFINE ENTRIES OF TABLE 742
LOOP ON ENTRIES OF TABLE 742
USE FUNCTIONS TO CALCULATE THE PROOFER CMY VALUES FROM PRINTER CMY VALUES OF TABLE 742
OUTPUT AN ENTRY IN TABLE 742
END LOOP TABLE 742 ENTRIES
END

FIG.11

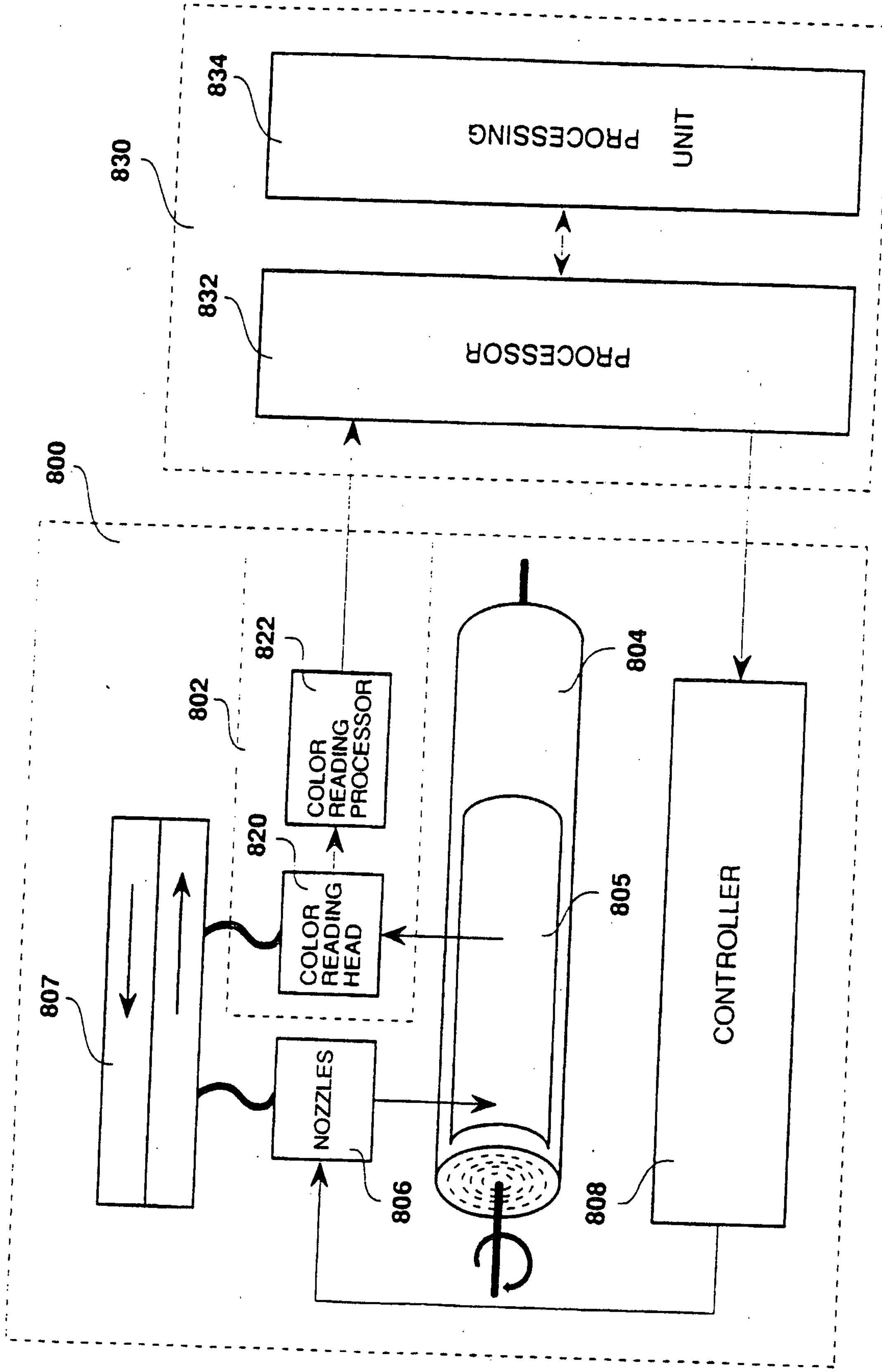

830
834
PROCESSING
UNIT
832
PROCESSOR
800
822
COLOR
READING
PROCESSOR
802
820
COLOR
READING
HEAD
804
805
807
NOZZLES
806
CONTROLLER
808

FIG. 12

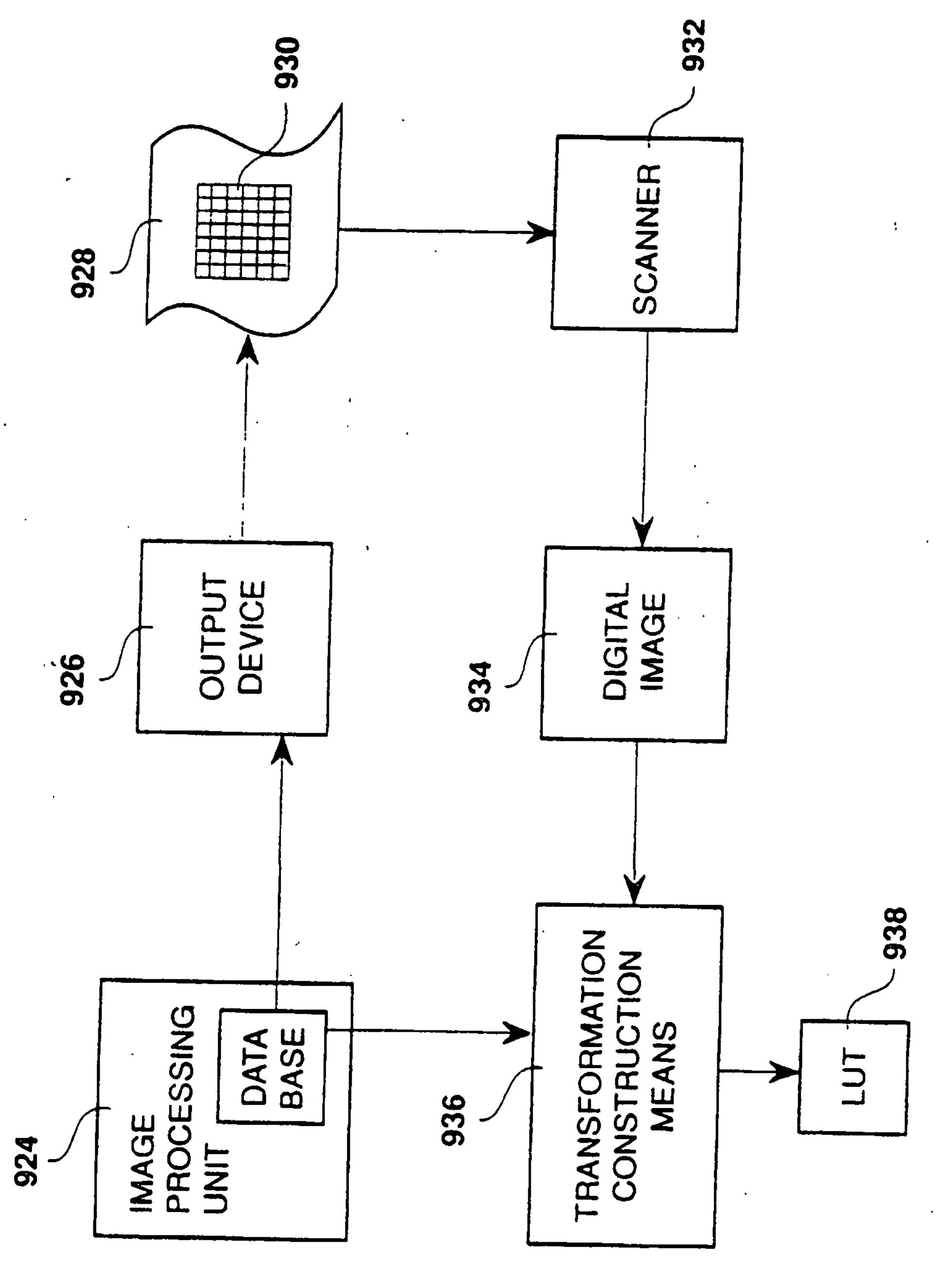

928
930
932
SCANNER
926
OUTPUT DEVICE
934
DIGITAL IMAGE
924
IMAGE PROCESSING UNIT
DATA BASE
936
TRANSFORMATION CONSTRUCTION MEANS
938
LUT

FIG. 13

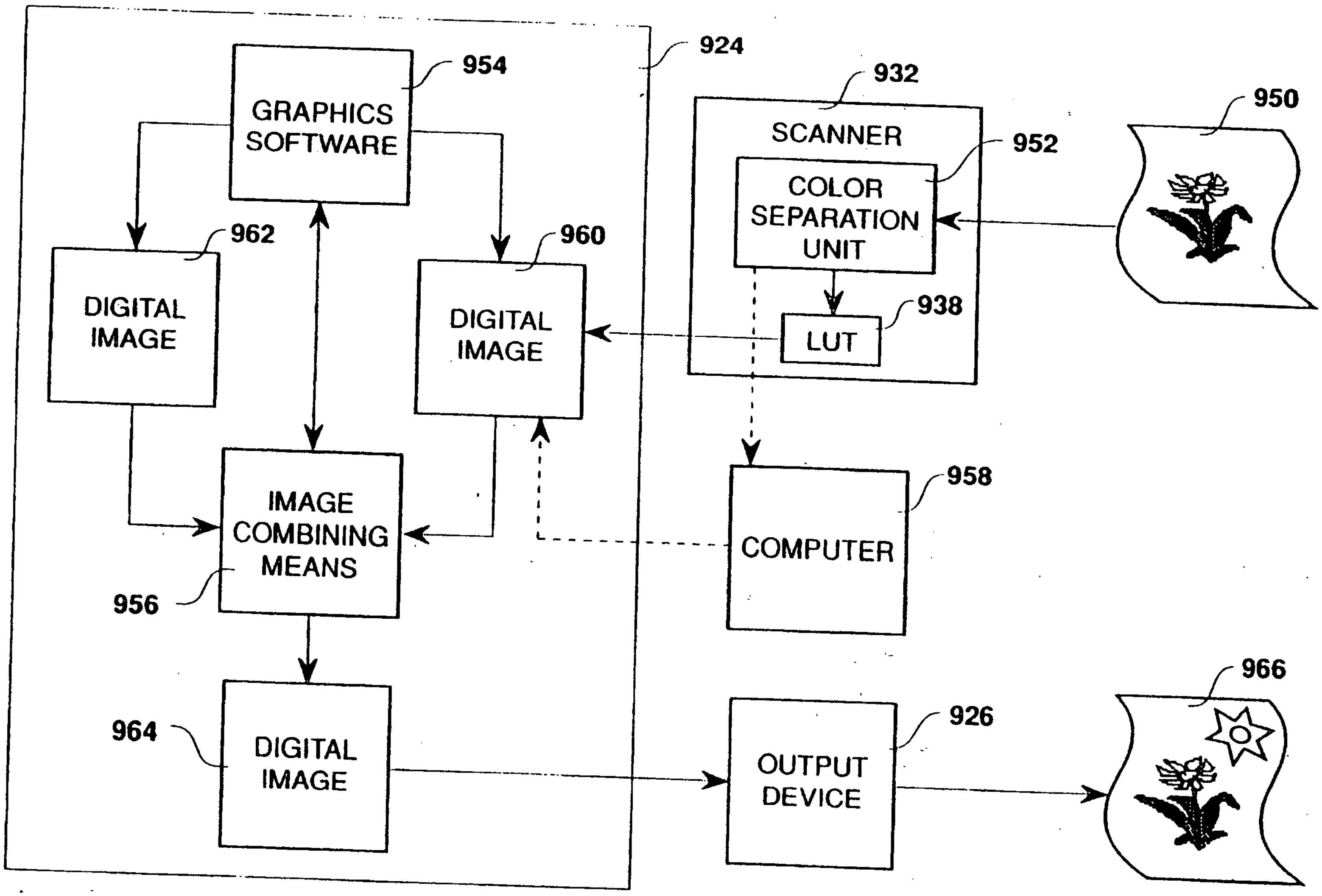

924
954
GRAPHICS SOFTWARE
962
DIGITAL IMAGE
960
DIGITAL IMAGE
IMAGE COMBINING MEANS
956
964
DIGITAL IMAGE
932
SCANNER
952
COLOR SEPARATION UNIT
938
LUT
950
958
COMPUTER
926
OUTPUT DEVICE
966

FIG. 14

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 475 554 A3**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: **91300904.9**

(51) Int. Cl.5: **H04N 1/46**

(22) Date of filing: **04.02.91**

(30) Priority: **27.12.90 IL 96816**
30.12.90 IL 96829
15.01.91 IL 96955
05.02.90 IL 93274

(43) Date of publication of application:
18.03.92 Bulletin 92/12

(84) Designated Contracting States:
AT BE CH DE DK ES FR GB GR IT LI LU NL SE

(88) Date of deferred publication of the search report:
23.06.93 Bulletin 93/25

(71) Applicant: **SCITEX CORPORATION LTD.**
7 Hamada Street
Herzliya 46103(IL)

(72) Inventor: **Smilansky, Zeev**
7 Beeri Street
Rehovot(IL)
Inventor: **Melman, Haim Zvi**
3 Hagai Street
Kfar Saba(IL)
Inventor: **Bresler, Yoav**
24 Refidim Street
Tel Aviv(IL)
Inventor: **Litvak, Matty J.**
42 Ha'atzmaut Street
Petach Tikva(IL)
Inventor: **Kreitman, Haim**
14/3 Hagra Street
Kfar Saba(IL)
Inventor: **Yad-Shalom, Itai**
6 Hagilad Street
Kfar Saba(IL)

(74) Representative: **Hillier, Peter et al**
Reginald W. Barker & Co., 13, Charterhouse Square
London, EC1M 6BA (GB)

(54) Apparatus and method for colour calibration.

(57) There is disclosed technique and apparatus for calibrating a color processing device. The technique includes the steps of comparing a first digital representation of a colored image with a second digital representation thereof and employing at least the transformation to control operation of the color processing device to be calibrated. The first digital representation defines a plurality of first non-scalar color values and the second digital representation defines a plurality of second non-scalar color values corresponding to the plurality of the first non-scalar color values, thereby to provide a transformation pairing each individual one of the first non-scalar color values with a value relatively close to the corresponding one of the second non-scalar color values. The color processing device may be calibrated generally without reference to human aesthetic judgement.

There is additionally provided a method and apparatus for transforming an element of a domain of a first color printing device to an element of a domain of a second color printing device. The method comprises the steps of providing a first transformation from a first digital representation of a colored image to a second digital representation thereof and a second transformation from a third digital representation of a colored image to a fourth digital representation thereof and comparing the first transformation with the second transformation. The second transformation corresponds to the second color printing device, the first transformation corresponds to the first color printing device and the second and fourth digital representations are defined within a single color space.

EP 0 475 554 A3

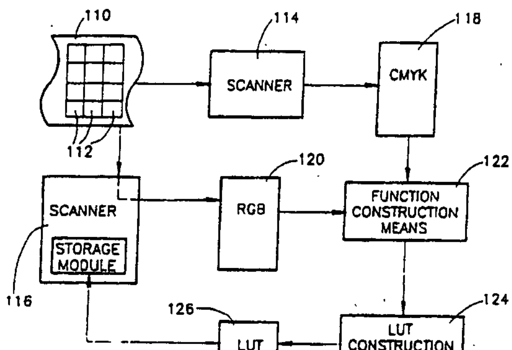

110
114
118
SCANNER
CMYK
112
120
122
SCANNER
RGB
FUNCTION CONSTRUCTION MEANS
STORAGE MODULE
116
126
124
LUT
LUT CONSTRUCTION MEANS

FIG.1A

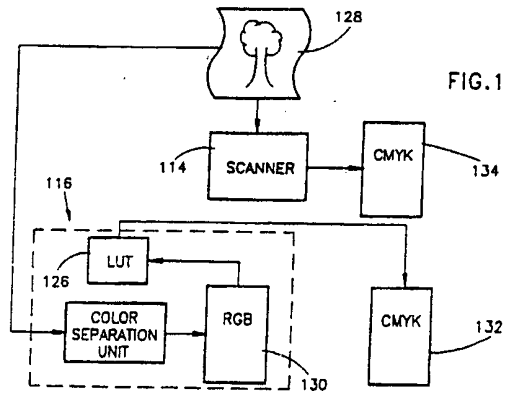

128
114
SCANNER
CMYK
134
116
LUT
126
COLOR SEPARATION UNIT
RGB
CMYK
132
130

FIG.1B

European Patent Office

EUROPEAN SEARCH REPORT

Application Number

EP 91 30 0904

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. Cl.5)
D,X Y A	EP-A-0 144 188 (XEROX) * page 2, paragraph 2 - page 6, paragraph 4 *	1-5, 10-12 7,9 6,8	H04N1/46
D,Y	ACM TRANSACTIONS ON GRAPHICS vol. 7, no. 4, October 1988, pages 249 - 292 M. C. STONE ET AL. 'Color Gamut Mapping and the Printing of Digital Color Images' * page 255, paragraph 2.4 * * page 261, paragraph 332 *	7,9	
A	PATENT ABSTRACTS OF JAPAN vol. 13, no. 327 (M-854)24 July 1989 & JP-A-11 10 154 (MATSUSHITA) 26 April 1989 * abstract *	-	
A	& PATENT ABSTRACTS OF JAPAN vol. 13, no. 364 (E-805)14 August 1989 & JP-A-11 20 965 (MATSUSHITA) 12 May 1989 * abstract *	-	TECHNICAL FIELDS SEARCHED (Int. Cl.5) H04N
A	& PATENT ABSTRACTS OF JAPAN vol. 14, no. 168 (E-912)30 March 1990 & JP-A-20 23 776 (MATSUSHITA) 25 January 1990 * abstract *	-	
P,A	& US-A-4 929 978 (KANAMORI ET AL.) 29 May 1990	-	
A	FR-A-2 455 307 (DAINIPPON SCREEN SEIZO K.K.)	-	

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	08 APRIL 1993	DE ROECK A.F.A.

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

This Page Blank (uspto)